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
described in this document. The furnishing of this document does not give you any license to these
patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation,
500 Columbus Avenue, Thornwood, NY, 10594, USA.

Programming Interface Information

This guide describes the program administration tasks involved in using TME 10
Information/Management.

This guide also documents General-use Programming Interface and Associated Guidance Information
provided by Information/Management.

General-use programming interfaces allow the customer to write programs that obtain the services of
Information/Management.

General-use Programming Interface and Associated Guidance Information is identified where it occurs,
either by an introductory statement to a chapter or section or by the following marking: General-use
Programming Interface.
Contents

Preface
Who Should Read This Guide ........................................... xiii
Related Publications .................................................... xiii
What This Guide Contains ............................................ xiv
Typeface Conventions ................................................... xvi
Contacting Customer Support ........................................ xvi

Defining a User Profile
Creating Default Profiles ............................................. 1-1
Defining a Default Profile ............................................. 1-2
Controlling User Profiles ............................................. 1-2
  Prompting the User with a Stored Response Chain ............. 1-3
  Changing Validation of Data-Entry Fields ....................... 1-3
  Deleting Profile Data-Entry Fields .............................. 1-4
  Adding User-Profile Variable Fields ............................. 1-4
Using a Startup TSP to Update Profiles ......................... 1-12
Setting Values for the PDF Editor ................................. 1-13
  Specifying an Invocation Macro for the PDF Editor .......... 1-14
Customizing the LRECL for PDF ..................................... 1-14
Preserving the Audit Trail for Freeform Text .................... 1-15

Defining Privilege Classes
Assigning Privilege Classes ......................................... 2-1
Categories of Users ................................................... 2-1
  Master Privilege Class ........................................... 2-2
  Privilege Class Structure ....................................... 2-2
Understanding Authorities ........................................... 2-5
  Function Level Authorities ..................................... 2-6
  Field Level Authorities ........................................ 2-7
  PMF Authorities .................................................. 2-8
Database Administration Authority ............................... 2-10
Universal Partition Access Authority ............................. 2-11
Assigning Authorities ............................................... 2-11
Example of Authority Assignments for Problem Management Users .................................................. 2-12
Deciding Whether to Use Record Ownership .................... 2-13
Managing Privilege Classes ............................................. 2-13
  Defining an Individual Privilege Class .......................... 2-14
  Defining a Group Privilege Class ................................. 2-27
  Determining All Privilege Class Names ......................... 2-34
  Displaying Authorities for a Privilege Class .................. 2-39
  Deleting One Member of a Privilege Class ...................... 2-42
  Deleting a Privilege Class ......................................... 2-44

Understanding Your Information/Management Database
The Databases .................................................................. 3-1
  Structured Description Data Set (SDDS) ......................... 3-2
  Structured Description Index Data Set (SDIDS) ............... 3-2
  Structured Description Log Data Set (SDLDS) .................. 3-2
Your Database Options .................................................. 3-4
  Multiple-Cluster SDDS ................................................. 3-5
  SDDS Key Length .................................................. 3-5
  Multiple-Cluster SDIDS ............................................... 3-5
  SDIDS Key Length .................................................. 3-6
Determining Your Backup/Recovery Approach ..................... 3-6
Reuse of SDDS Position Numbers (Root VSAM Keys) .......... 3-8
Creating Database Sorted by Record Number ...................... 3-8
  Planning .................................................................. 3-8
  Procedure .......................................................... 3-9

Logical Database Partitioning
Primary Partition Name .................................................. 4-1
Owning Partition Name .................................................. 4-1
Secondary Partition List .................................................. 4-2
Authorizing a Privilege Class for Logical Database Partitions 4-2
Privilege Class Selection .............................................. 4-7
Universal Partition Access Authority ............................... 4-8
Global Partition .......................................................... 4-13
Sharing Partition List .................................................... 4-13
Logical Database Partitioning Considerations .................... 4-17
  Unique RNIDs by Partition ......................................... 4-17
  Record Searches .................................................... 4-17
  Copying Records between Partitions ............................. 4-18
  Record Ownership ................................................... 4-18
Commands in a Logical Partition Environment .................... 4-18
  The COPY, DISPLAY, PRINT, DELETE, and UPDATE Commands 4-18
  The GLOSSARY Command ......................................... 4-19
  The ORDER Command .............................................. 4-19
  The STATISTICS Command ...................................... 4-19
Migrating Existing Records to a Logical Partition ............... 4-19
PMF Considerations ..................................................... 4-20
  Predefined Validation Patterns ................................... 4-20
Archiver Considerations .................................................. 4-21
Customizing the Enhanced Panel Style

Information/Management Panel Styles ......................................................... 5-1
  Standard Panel Style ............................................................................. 5-2
  Enhanced Panel Style ............................................................................ 5-3
Understanding the ISPF Panels ................................................................. 5-5
  ISPF Tables .......................................................................................... 5-6
  Variables for Automating Window Selection ......................................... 5-6
Enabling the Enhanced Panel Style for ISPF GUI ........................................... 5-7
Changing the Default ISPF Panel ................................................................. 5-8
Changing the Code for an ISPF Panel ......................................................... 5-8
  Adding an Action to the Action Bar ......................................................... 5-9
  Changing the Action Bar Action or Pull-Down Choice ............................ 5-15
Using DTL to Change an Action Bar ............................................................ 5-16
Changing Keylists ...................................................................................... 5-17
  Adding Accelerator Keys and Separator Bars ........................................ 5-20
Creating an Enhanced Panel Style Window ................................................ 5-21
Using the Standard Panel Style ................................................................... 5-21

Performing Database Administration

Performing DBCLEANUP ............................................................................. 6-2
Performing CHECK IN .............................................................................. 6-4

Using Notification Management

Immediate Notification Using TSXs and TCP/IP .............................................. 7-2
  Notification Processing with TSXs .......................................................... 7-2
  Immediate Notification with TSXs ............................................................ 7-3
  Enabling Immediate Notification with TSXs Using TCP/IP SMTP ............... 7-4
  Creating the USERS Record .................................................................... 7-10
Immediate Notification Using TSPs and TSO Send ......................................... 7-18
  Enabling Immediate Notification ............................................................ 7-19
  Creating the USERS Record .................................................................... 7-22
  Changing the Notification Message .......................................................... 7-28
  Disabling Immediate Notification .............................................................. 7-47
Understanding Problem Escalation .............................................................. 7-49
  Implementing Problem Escalation .............................................................. 7-50
  Understanding the Escalation Process ....................................................... 7-50
  The Escalation TSPs and TSXs ................................................................. 7-52
  Creating Rules Records ........................................................................... 7-53
  Understanding the Escalation Search Argument ....................................... 7-56
  Escalating Problems ................................................................................. 7-58
  Rules Record 1 ........................................................................................ 7-58
  Rules Record 2 ........................................................................................ 7-59
  Sending the Alert Message ....................................................................... 7-60
Starting Escalation Processing .................................................................... 7-60
  Creating the JCL for the Escalation Batch Job .......................................... 7-60
Tailoring the Escalation Facility ................................................................... 7-61
  Changing the Escalation Criteria ............................................................... 7-62
## Contents

Changing the Escalation Command Template ........................................ 7-62
Changing the Escalation Message ...................................................... 7-63
Using the BLGESCX6 TSX with TCP/IP SMTP .................................. 7-64
Notification Management User Exits ............................................... 7-65
  The Escalation Control Block ..................................................... 7-65

### Defining User Line Commands

Defining User Line Commands for Search Results Lists ......................... 8-1
  Defining User Line Commands and Where They Are Valid .................. 8-2
Using TSP and TSX Aliases .......................................................... 8-5
Creating the ALIAS Record ........................................................... 8-6
Processing User Line Commands ..................................................... 8-10
  Example User Exit ................................................................. 8-12
  TSP and TSX Considerations .................................................... 8-12

### Defining Command Aliases and Authorization

Creating the COMMAND Record ....................................................... 9-2
Usage Notes .................................................................................. 9-6

### Setting Up Stored Response Chains

Assessing Needs for SRCs ............................................................... 10-1
Planning SRCs ............................................................................. 10-1
Modifying and Controlling SRCs ..................................................... 10-3
  Rules for Modifying SRCs .......................................................... 10-3
Deleting SRCs .............................................................................. 10-3
Controlling SRCs ........................................................................ 10-4

### Propagating Information/Management Data

Understanding Data Propagation ...................................................... 11-1
  Mode of Propagation .................................................................. 11-4
Choosing the Data to Be Propagated ............................................. 11-5
Ensuring the Integrity of the Propagated Data .................................. 11-5
Frequency of Data Propagation ..................................................... 11-5
Using Propagated Data ................................................................. 11-6
Automatic Log Save Facility ......................................................... 11-7
  Automatic Log Save Send Function .............................................. 11-7
  Automatic Log Save Receive Function ......................................... 11-8
DB2 Extract Facility ....................................................................... 11-9
  DB2 Extract Send Utility .............................................................. 11-9
  DB2 Update Utility ..................................................................... 11-10
Considerations for Using Both Data Propagation Facilities ............... 11-12
Performance and Environmental Considerations ............................ 11-14
Understanding the LOGSAVE Record ............................................. 11-15
  Fields on Panel BLG0Z500 .......................................................... 11-16
  Fields on Panel BLG0Z510 .......................................................... 11-18
  Fields on Panel BLG0Z510 .......................................................... 11-19
  Fields on Panel BLG0Z530 .......................................................... 11-20
Contents

Maintaining Synchronization .................................................. 11-22

Propagating to an Information/Management Database

Implementing the Automatic Log Save Facility ........................................ 12-2
Understanding Automatic Log Save Send Processing ............................... 12-3
Understanding Automatic Log Save Receive Processing ............................. 12-4
Setting Up the Automatic Log Save Send Function ..................................... 12-6
  Creating a LOGSAVE Record for a Send Database .................................. 12-7
  Creating a GDG ........................................................................ 12-14
  Preparing the JCL .................................................................... 12-15
  Modifying the BLGDUMP1 TSP ..................................................... 12-20
Setting Up the Automatic Log Save Receive Function ................................. 12-21
  Copying the LOGSAVE Record into the Receive Database ......................... 12-22
  Updating the LOGSAVE Record for Receive ......................................... 12-24
  Filtering Records ...................................................................... 12-32
  Copying the JCL ....................................................................... 12-32
Starting Normal Send and Receive Processing ........................................... 12-32
Stopping Send and Receive Processing .................................................... 12-32
Using the Automatic Log Save Facility for Automatic Backup ..................... 12-33
Disabling Automatically Starting the Next Send ......................................... 12-33
Deleting a LOGSAVE Record .................................................................. 12-34
  Deleting a Send LOGSAVE Record .................................................. 12-34
  Deleting a Receive LOGSAVE Record .............................................. 12-35
Recovering from Errors ....................................................................... 12-35
  Error during Send or Receive Processing ............................................ 12-36
  Preparing BLMSASDE .................................................................. 12-36
  Restoring Synchronization ............................................................. 12-37

Propagating to a DB2 Database

Implementing the DB2 Extract Facility .................................................... 13-1
Understanding DB2 Extract Send Processing ............................................ 13-2
Understanding DB2 Update Processing .................................................... 13-4
Setting Up the DB2 Extract Facility .......................................................... 13-5
  Building Relational Data Mapping Tables ............................................ 13-5
  Defining a LOGSAVE Record for DB2 Extract Processing ....................... 13-13
  Establishing the DB2 Environment .................................................... 13-18
  Modifying DSNTEP2 .................................................................... 13-21
  Creating GDGs ........................................................................... 13-22
  Preparing the JCL ...................................................................... 13-23
  Modifying the DB2 Extract TSP ....................................................... 13-29
Starting Send and Update Processing ....................................................... 13-31
Example of Propagating Problem Data to DB2 ........................................ 13-32
Stopping the DB2 Extract Facility ............................................................ 13-34
Disabling Automatically Starting the Next Send ....................................... 13-34
Propagating Data to Both Information/Management and DB2 .................... 13-35
  Changing the TSP ....................................................................... 13-36
  Changing the JCL for the Automatic Log Save Facility ......................... 13-37

ix
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing the JCL for the DB2 Extract Facility</td>
<td>13-37</td>
</tr>
<tr>
<td>Starting Propagation to Both</td>
<td>13-37</td>
</tr>
<tr>
<td>DSNTEP2 Messages</td>
<td>13-38</td>
</tr>
<tr>
<td>Recovering from Errors</td>
<td>13-38</td>
</tr>
<tr>
<td>Recovering from a BLGDBXM Failure</td>
<td>13-39</td>
</tr>
<tr>
<td>Recovering When There Is a Receive Error</td>
<td>13-40</td>
</tr>
<tr>
<td>Recovering When the SDLDS Is Full</td>
<td>13-40</td>
</tr>
<tr>
<td>Resetting Synchronization</td>
<td>13-41</td>
</tr>
</tbody>
</table>

#### Data Propagation User Exits

User Exits ........................................................................ 14-1

#### The Information/Management Archiver

Functional Overview .................................................................. 15-1
Input Data Sets ........................................................................ 15-5
Output Data Sets ....................................................................... 15-5
Return Codes ............................................................................ 15-5
Input Parameters ...................................................................... 15-6
Required Parameters .................................................................. 15-15
BLGINREC Sample ...................................................................... 15-17
BLGARCP Samples ..................................................................... 15-18
Sample 1 .................................................................................. 15-18
Sample 2 .................................................................................. 15-19
Sample 3 .................................................................................. 15-20
Sample 4 .................................................................................. 15-21
Sample 5 .................................................................................. 15-22
Parent/Child Relationships ...................................................... 15-23
IBM-defined Relationships ...................................................... 15-23
User-defined Relationships ..................................................... 15-24
Sample RFT ............................................................................... 15-27
Additional Tailoring Considerations ......................................... 15-28
Performance Considerations .................................................... 15-28
Record Processing Considerations ........................................... 15-29
Panel Processing Considerations .............................................. 15-34
Logical Database Partition Considerations ................................ 15-35

#### Panel Lists

Information/Management Panels Arranged by Panel ID .................. A-1
Information/Management Panels Arranged by Panel Title ............... A-20

#### Training Records

Creating Training Records ......................................................... B-1

#### Your Organization’s Procedures

Getting Started ........................................................................ C-1
Numbering a Record .................................................................. C-2
Preface

This guide describes the program administration tasks involved in using TME 10 Information/Management Version 1.1 (hereafter referred to as Information/Management). This guide complements the Planning and Installation Guide and Reference and describes procedures involved in migrating to this release of Information/Management from previous versions.

There may be references in this publication to earlier versions of Information/Management (such as Information/Management Version 6.3, 5.1, 4.2, etc.). When Information/Management became a Tivoli product in 1997, its name changed to TME 10 Information/Management and its version number was reset to Version 1.1.

Who Should Read This Guide

This guide is intended for:

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

To use this guide effectively, you must understand how to use applications on your MVS/ESA operating system. You need to know how to use Information/Management, including how to create stored response chains (SRCs). Therefore, read the User’s Guide before you attempt to do any of the tasks in this book. To do some of the tasks described in this book, you must also know how to use the Panel Modification Facility (PMF) and Terminal Simulator Panels (TSPs). Refer to the Panel Modification Facility Guide and the Terminal Simulator Guide and Reference for more information.

Related Publications

The library for Information/Management Version 1.1 consists of these publications. For a description of each, see “The Information/Management Library” on page E-1.

- Application Program Interface Guide, SC34-4592-00
- Client Installation and User’s Guide, SC34-4593-00
- Diagnosis Guide, SC34-4594-00
- General Information, GC34-4595-00
Preface

Integration Facility Guide, SC34-4596-00
Licensed Program Specification, GC34-4597-00
Master Index, Glossary, and Bibliography, SC34-4598-00
Messages and Codes, GC34-4599-00
NetView and TME 10 Interfaces Reference, SC34-4613-00
Operation and Maintenance Reference, SC34-4601-00
Panel Modification Facility Guide, SC34-4602-00
Planning and Installation Guide and Reference, SC34-4603-00
Problem, Change, and Configuration Management, SC34-4604-00
Program Administration Guide and Reference, SC34-4605-00
Reference Summary, SC34-4607-00
Report Format Facility Guide and Reference, SC34-4608-00
Terminal Simulator Guide and Reference, SC34-4609-00
User’s Guide, SC34-4610-00
World Wide Web Interface Guide, SC34-4570-00

A subscription to the Information/Management customer newsletter, The Structured Word, is available to you free of charge. Send your complete mailing address to:

Tivoli Systems
The Structured Word
Dept. X7GA/Bldg. 503
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

What This Guide Contains

The following chapters provide information about administering Information/Management databases:

- “Defining a User Profile” on page 1-1 tells you how to define and set up default user profiles.
- “Defining Privilege Classes” on page 2-1 tells you how to create privilege classes.
- “Logical Database Partitioning” on page 4-1 discusses how you can organize your data into multiple “logical” partitions from a single physical database.
- “Customizing the Enhanced Panel Style” on page 5-1 tells you how to customize the enhanced panel style.
- “Performing Database Administration” on page 6-1 tells you how to use DBCLEANUP and CHECK IN.
- “Using Notification Management” on page 7-1 tells you how to notify people of the status of problem, change, or activity records, including how to use REXX TSP (TSXs) and TCP/IP for notification. In addition, it describes how to create the USERS and the RULES records and tailoring the escalation facility.
“Defining User Line Commands” on page 8-1 tells you how to create user line commands.

“Defining Command Aliases and Authorization” on page 9-1 tells you how to customize Information/Management commands and restrict the usage of commands by privilege class.

“Setting Up Stored Response Chains” on page 10-1 tells you how to set up SRCs.

The following chapters provide information about propagating Information/Management data:

“Propagating Information/Management Data” on page 11-1 discusses data propagation and the Information/Management facilities for propagating data.

“Propagating to an Information/Management Database” on page 12-1 tells you how to set up the Automatic Log Save Facility to propagate data to another Information/Management database.

“Propagating to a DB2 Database” on page 13-1 tells you how to set up the DB2 Extract Facility to propagate data to a DB2 database.

“Data Propagation User Exits” on page 14-1 lists the return and reason codes for the data propagation user exits.

The following chapter provides information about archiving Information/Management data:

“The Information/Management Archiver” on page 15-1 discusses the Archiver function that enables you to move records from one Information/Management database to another.

The following appendixes provide reference information:

“Panel Lists” on page A-1 lists the entry, display, and inquiry panels that are shipped with Information/Management.

“Training Records” on page B-1 contains sample training records you can use to create a set of training records for beginning users to use in conjunction with the step-by-step exercises in the User’s Guide. You must create training records before beginning users can work with exercises in the User’s Guide.

“Your Organization’s Procedures” on page C-1 provides a template for documenting how your organization performs certain tasks (for example, entering data and numbering records) with the Information/Management program. You can use this template to provide operating procedures for all users in your organization to follow.
How Information Is Presented in This Book

The panels as presented in this book are not meant to be exact replicas of the way a panel might appear on the screen. The information on the panels is correct, but the spacing is not always exact.

With Information/Management, you may see changes in the way Information/Management panels are displayed. 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.

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. For more information, see the glossary.

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**.

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

  **Monospace**
  Code examples appear in monospace font.

The panels as presented in this book are not meant to be exact replicas of the way a panel might appear on the screen. The information on the panels is correct, but the spacing is not always exact.

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.

Contacting Customer Support

We are very interested in hearing from you about your experience with the products and documentation in the TME 10 Information/Management environment. We welcome your suggestions for improvements.

To contact Tivoli for support, call IBM Support Services at any time. IBM Support Services provides telephone assistance for problem diagnosis and
resolution in the United States and Puerto Rico. The number to call is (800) 237-5511. Outside the United States and Puerto Rico, contact your country’s local IBM support organization.

You can fax comments on this publication to (919) 254-0206. When you send information to Tivoli, you grant Tivoli a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

You can also order publications by phone or fax. IBM Software Manufacturing Solutions takes publication orders between 8:30 a.m. and 7:00 p.m. eastern standard time (EST). The phone number is (800) 879-2755. The fax number is (800) 445-9269 (attention fax department).
Defining a User Profile

This chapter suggests ways that the program administrator can:

- Create a default profile
- Define a default profile
- Control user profiles
- Set values for the PDF editor

As the program administrator, you can create a default profile for all users. Your users can modify the default profile to meet their needs. By creating a default profile, you ensure that certain values are initially entered properly. Use the Panel Modification Facility (PMF) to modify profile panels to control what values users can and cannot alter. Refer to the Panel Modification Facility Guide.

Creating Default Profiles

A user’s profile determines some of the ways Information/Management operates by providing default data values during sessions. If the user has not set up a user profile, Information/Management uses one you can set up as the default. You can control both the contents of the default profile and what individual users can define in their own user profiles. You must, of course, establish your controls before any other users sign on to Information/Management.

When Information/Management uses a default profile, it first searches for the member BLG0PROF in the partitioned data set (PDS) allocated to the user’s ISPPROF DD statement in Interactive System Productivity Facility (ISPF). BLG0PROF in the ISPPROF DDName concatenation is the user’s own ISPF profile for the Information/Management application. If BLG0PROF is not there, ISPF searches for the same member (BLG0PROF) in the PDS allocated to the ISPTLIB DD statement in ISPF. BLG0PROF in the ISPTLIB DDName concatenation is the Information/Management profile. If member BLG0PROF is found in the data set, the profile is read into the ISPF profile variable pool and Information/Management uses it for the session. If member BLG0PROF is not found, Information/Management builds its own profile using internal default values and places it in the ISPF profile variable pool.

When the user ends the current Information/Management session, ISPF writes the contents of the Information/Management profile variable pool to the user’s profile data set as the BLG0PROF member so that it is available for the next session.
Defining a Default Profile

By ensuring that the Information/Management profile member of the ISPTLIB PDS exists, you can put your own default values into effect to override the internally defined defaults. In addition, because the ISPF variable pool contains PF key settings as well, you can set default PF keys for an Information/Management session.

Before creating a default profile for your Information/Management users, consider the following questions:

■ What defaults do you want to provide your users?
■ Do different groups of users require different defaults, or can one value for a given field serve all users?

To define a default profile for new users to use the first time they sign on to Information/Management, follow these steps:

1. Start Information/Management and select Profile from the Primary Options Menu or enter the PROFILE command.
2. Update the profile by setting the default values you want to include in the default profile.
3. After you finish, select Permanent profile end from the Profile Summary panel, or enter the END command to permanently update the profile.
4. Issue the ISPF KEYS command and set the PF key values to those you want to include in the default profile.
5. Enter END to complete the ISPF KEYS command.
6. Enter QUIT to exit the Information/Management session and to write the profile to your profile data set.
7. Copy the BLG0PROF member from your profile data set into ISPF’s system profile data set (allocated to the ISPTLIB DD statement).

Controlling User Profiles

You can ensure that users do not change certain values in the default profile you have established, or, when they do change the values, that they enter proper alternatives by one of the following methods:

■ Use a startup stored response chain (SRC) to prompt the user for correct responses. See “Prompting the User with a Stored Response Chain” on page 1-3 for more information.
■ Modify the validation process for certain data fields in the default profile. See “Changing Validation of Data-Entry Fields” on page 1-3 for more information.
■ Remove data fields from the panel. See “Deleting Profile Data-Entry Fields” on page 1-4 for information on deleting data fields.
You can also add your own data fields to the profile. See “Adding User-Profile Variable Fields” on page 1-4 for information on adding data fields.

You can also use a startup TSP to change user profiles and ensure that they contain the correct values. This startup TSP can also be used to make the same change to every user’s profile. BLGTPSET is an example of a TSP that you can use for this purpose. See “Using a Startup TSP to Update Profiles” on page 1-12 for information on TSP BLGTPSET.

Prompting the User with a Stored Response Chain
For those fields that pertain to the individual user (user’s name, department, and telephone number), you can create a startup SRC that selects Profile from the Primary Options Menu and automatically prompts the new user for these items. Assuming that you assign all users to privilege classes, and that the users know the names of those privilege classes the first time they sign on to Information/Management, you can also prompt for their invocation privilege class (a session control default).

To ensure that new users are prompted for this information only the first time they sign on to Information/Management, have the startup SRC replace its own name in the Invocation SRC field in the user profile with blanks; or, have the startup SRC update the Invocation SRC field with the name of the SRC that processes each time the user logs on. Also, design the startup SRC to select Permanent profile end, which forces the SRC to end.

If you specify this SRC’s name in the Invocation SRC field of the default profile, new users signing on to Information/Management for the first time are automatically prompted for this data and their profiles are permanently updated with their responses. See “Setting Up Stored Response Chains” on page 10-1 for more information.

Changing Validation of Data-Entry Fields
You can also control how the user modifies the default profile by changing the validation process for the data collected in the profile fields. Whenever a user enters data into a data-entry field on the profile panels, Information/Management calls program exit BLG01439 to compare the data to a set of acceptable values. Any value that does not match one of the acceptable values is ignored, and an error message is issued. Refer to the Panel Modification Facility Guide for more information about the BLG01439 program exit.

You can modify the field validations to further limit the user’s choice of values within the values that are already defined. For example, you could allow only responses of command or data for the command detection field and eliminate prompt as a response. You cannot, however, change the prompt response to ask; profile validation does not process this value.

By modifying the validation of certain data fields, you can ensure that the user does not change the data in the default profile to a value that is not appropriate in your organization. However, you do not want to be too restrictive, because you could limit the ease-of-use that the profile function provides.
Refer to the *Panel Modification Facility Guide* for information about changing validation of data-entry fields.

**Deleting Profile Data-Entry Fields**
You can delete any field from the profile data-entry panels with PMF. When you eliminate certain fields from the profile panels, users cannot alter the default profile data associated with those deleted fields. This method ensures that all users use profile values appropriate for your organization.

Refer to the *Panel Modification Facility Guide* for information about modifying fields on panels.

**Adding User-Profile Variable Fields**
You can define as many user-profile variable fields as you want. A *user-profile variable field* is a field in the user’s profile that can have the value in it extracted and used when the user enters the equal sign in a data-entry field. Information/Management has this capability built in for the user’s name, department, and phone number. You might have others you want to use.

You cannot add user-profile variables to the existing user-profile data-entry panels, but Information/Management provides a special ISPF panel for this purpose. The Profile Summary panel selection *User-defined profile values* enables the user to enter data in the new variables and use them like other profile fields, including using the equal sign function to enter the data on data-entry panels. To define your own user-profile variables, modify the ISPF panel BLGPVARS.

First, find where panel BLGPVARS resides in your system. The Information/Management installation instructions tell you to have this panel reside in a data set that is in your ISPPLIB concatenation. You must update this panel in the way you usually update data sets. Because you cannot update this panel using PMF, this example uses the PDF editor. For more information on updating ISPF panels, refer to *ISPF/PDF Edit and Edit Macros ISPF/PDF Guide and Reference*, and *ISPF Dialog Management Guide and Reference*.

In the following example, the panel is in the INFO.SPECIAL.PANELS data set. Suppose that you want to replace the standard Information/Management field for user’s name with one of your own. You also want a field for the user’s location in the user profile. The following example shows step-by-step how to replace the user’s name field, and how to create a user’s location field.
Defining a User Profile

Controlling User Profiles

This is how panel BLGPVARS appears on your screen when you are in PDF edit mode.

This is the way the panel is shipped with Information/Management. The underscore character (_) represents the beginning of an entry field. The percent (%) and plus (+) characters represent the beginning of protected fields. Lines 1-20 contain the visible part of the panel. Lines 21-23 contain statements that do panel processing.

```
EDIT ---- INFO.SPECIAL.PANELS(BLGPVARS)---- 01.00 -------- COLUMNS 001 072
COMMAND ===> SCROLL ===> PAGE
****** ********************************************* TOP OF DATA *********************************************
000001 %---------------------------------USER DEFINED PROFILE VARIABLES-----------------------------
000002 +
000003 *********************************************
000004 ****
000005 **** This panel should be modified to suit your installation's needs.
000006 **** A single variable is shown below to help you get started.
000007 ****
000008 *********************************************
000009 +
000010 +
000011 +
000012 + User Variable 1 %====>_VAR1
000013 +
000014 +
000015 +
000016 +
000017 +
000018 +
000019 % PRESS ENTER TO RETURN TO PROFILE SUMMARY PANEL
000020 *********************************************
000021 )PROC
000022 VPUT (VAR1) PROFILE
000023 )END
****** ********************************************* BOTTOM OF DATA *********************************************
```
No matter what other changes you want to make to this panel, you probably want to remove the instructions from it. Simply deleting those lines causes the final panel to be shorter. To keep the standard panel length, remove just the instruction text and leave the + attribute character for each line as shown. For the rest of this example, the lines are deleted so that all of the processing lines of the panel can be seen on one screen. Because you probably want to keep the lines, your panel processing lines will spill over to a second editing screen.

```
EDIT ---- INFO.SPECIAL.PANELS(BLGPVARS)---- 01.00 -------- COLUMN 001 072
COMMAND ==> SCROLL ==> PAGE

******* *************************** TOP OF DATA ***************************
000001 %-----------------USER DEFINED PROFILE VARIABLES-------------------
000002 +
000003 +
000004 +
000005 +
000006 +
000007 +
000008 +
000009 +
000010 +
000011 +
000012 +
000013 +
000014 +
000015 +
000016 +
000017 +
000018 +
000019 % PRESS ENTER TO RETURN TO PROFILE SUMMARY PANEL
000020 %-------------------------------------------------------------------
000021 )PROC
000022 VPUT (VAR1) PROFILE
000023 )END

User Variable 1 %===>_VAR1

******* *************************** BOTTOM OF DATA ***************************
```
To identify a user-defined field, create an ISPF variable on this panel. Decide on the variable’s name and its length in characters. To avoid confusion, give every variable a unique name. Eight-character names offer plenty of chances for name uniqueness. This example uses 8-character names with an arbitrary naming convention that requires that all names start with the characters BLG.

Identify a field for the user’s name. Type over the existing information for User Variable 1, making sure to retain the attribute characters. Beginning with the character after the underscore, count out spaces equal to the maximum number of characters you want to allow for the length of the name. Put a plus sign to indicate the end of the field as shown. This example allows a length of 25 characters for the name. Change the characters after the underscore to BLGNAME, for the field My name, with our naming convention prefix of BLG. You just created an ISPF variable name.
Next identify a field for the user’s location. Put this information into line 8, rather than inserting a new line. (Because you did not delete the entire instruction line the way this example did, your line number for this step should be 13.) This keeps the current panel size. Call the ISPF variable for this field BLGUSLOC, and give the field a length of 15.

```
EDIT ---- INFO.SPECIAL.PANELS(BLPVARS)---- 01.00 ----------- COLUMNS 001 072
COMMAND ===> SCROLL ===> PAGE
****** ****************************************** TOP OF DATA ******************************************
000001 %-----------------USER DEFINED PROFILE VARIABLES------------------------
000002 +
000003 +
000004 +
000005 +
000006 +
000007 + My name %===>_BLGUNAME +
000008 + My location %===>_BLGUSLOC +
000009 +
000010 +
000011 +
000012 +
000013 +
000014 % PRESS ENTER TO RETURN TO PROFILE SUMMARY PANEL
000015 %-----------------------------------------------------------------------
000016 )PROC
000017 VPUT (VAR1) PROFILE
000018 )END
****** ****************************************** BOTTOM OF DATA ******************************************
```
You can make the profile fields you define required fields. This way you can ensure that certain information is always in the user’s profile. To make the fields required, change the processing portion of panel BLGVPVARS as shown here. Notice that you need to add a “verify” (VER) line (line number 17). Include the NB (Not Blank) parameter after the ISPF variable name to make the My name field a required field.

```
EDIT ---- INFO.SPECIAL.PANELS(BLGVPVARS)---- 01.00 --------- COLUMNS 001 072
COMMAND ===> SCROLL ===> PAGE
****** *********************** TOP OF DATA ***********************
000001 %-------------------USER DEFINED PROFILE VARIABLES------------------
000002 +
000003 +
000004 +
000005 +
000006 +
000007 + My name %===> _BLGUNAME +
000008 + My location %===> _BLGUSLOC +
000009 +
000010 +
000011 +
000012 +
000013 +
000014 % PRESS ENTER TO RETURN TO PROFILE SUMMARY PANEL
000015 %-----------------------------------------------------------
000016 )PROC
000017 VER (&BLGUNAME,NB);
000018 VPUT (VAR1) PROFILE
000019 )END
****** *********************** BOTTOM OF DATA ***********************
```
You can also prime the profile fields you define by changing the processing portion of the panel. This is useful when you create a field that is almost always going to contain the same information, for example, the location of a problem reporter.

To prime a profile field, add an INIT section to the panel processing. This example shows the My location field (BLGUSLOC) primed with myloc1. (See lines 16 and 17.) Also, you decide that only two entries are valid for the My location field, myloc1 and myloc2. You can make the processing portion of the panel verify that this field accepts only those two entries. Use a VER statement with a LIST parameter as shown in line 19.

```
EDIT ---- INFO.SPECIAL.PANELS(BLGPVARS)---- 01.00 001 072
COMMAND => SCROLL => PAGE
******** ***************************************** TOP OF DATA *****************************************
000001 %----------------------------USER DEFINED PROFILE VARIABLES----------------------------
000002 +
000003 +
000004 +
000005 +
000006 +
000007 +
My name %===>_BLGUNAME +
000008 +
My location %===>_BLGUSLOC +
000009 +
000010 +
000011 +
000012 +
000013 +
000014 %PRESS ENTER TO RETURN TO PROFILE SUMMARY PANEL
000015 %--------------------------------------------------------------------------------------
000016 )INIT
000017 IF (&BLGUSLOC = ' ') &BLGUSLOC='MYLOC1'; /* add location if not there */
000018 )PROC
000019 VER (&BLGUSLOC,LIST,MYLOC1,MYLOC2); /* list valid locations */
000020 VER (&BLGUNAME,NB);
000021 VPUT (VAR1) PROFILE
000022 )END
******** ***************************************** BOTTOM OF DATA *****************************************
```
If you want to make the **My location** field a required field, add the NB parameter to line 19 as well:

```plaintext
VER (&BLGUSLOC, NB, LIST, MYLOC1, MYLOC2);
```

Finally, save the information. Change the processing portion of the panel to save the new variables. Change line 21 to write the 2 variables you identified, BLGUNAME and BLGUSLOC, to the user’s profile.

To use the new profile variables in Information/Management, you must add a p-word to the dictionary for each of them. For this example, add **My location** field, which is identified by variable BLGUSLOC, and **My name** field, which is identified by variable BLGUNAME. You must add p-words to the dictionary with validation patterns of =BLGUSLOC and =BLGUNAME. These p-words must contain the prefix that is associated with the field each new user-defined profile variable is used for. For example, LOCC/ for location, and one entry for each name prefix, such as PERS/, PERA/, PERC/, PERR/, and so on.

When the prefix index is added as the first item in the list of validation patterns for an assisted-entry panel, the user can enter an equal sign into that field and retrieve the information that is stored for that field in the user’s profile. You must include the new variables on the panels that they are to be used on. Add their prefix indexes to the list of valid values for each panel that you want to use the equal sign on. Refer to the *Panel Modification Facility Guide* for information about adding entries to the dictionary and modifying fields on panels.

Now, when the user chooses option 12 from the Profile Summary panel, the user sees a panel with your user-defined profile variables. The blank lines that were
deleted during the example were replaced. The user can enter data here to include it in his user profile. Notice that My location field (BLGUSLOC) is primed with myloc1; however, the user can change it by entering new data in the field. The only other data that is accepted, however, is myloc2.

--- USER DEFINED PROFILE VARIABLES ---

My name
My location ====> MYLOC1

PRESS ENTER TO RETURN TO PROFILE SUMMARY PANEL

Using a Startup TSP to Update Profiles

If you want to make the same change to the profiles of all of your users, or ensure that their profiles contain the correct values, you can use a startup TSP. An example of a startup TSP that makes the same change to the profiles of all of your users is BLGTPSET. As this TSP is supplied, it updates the Bypass copyright? profile field to YES. The next time the user starts Information/Management, the proprietary information panel is bypassed.

You can run BLGTPSET as a startup TSP in either of the following ways:

- Use the TSP parameter on the ISPSTART command that starts Information/Management. For example:
  ISPSTART PGM(BLGINIT) PARM(SESS(/zerodot/zerodot) TSP(BLGTPSET))
- Use the IRC parameter on the ISPSTART command that starts Information/Management. For example:
  ISPSTART PGM(BLGINIT) PARM(SESS(/zerodot/zerodot) IRC(RUN BLGTPSET))

If you want to make other changes to all the profiles, you can use PMF to modify the TSP.

The logic of the TSP is as follows:

1. Call a user exit to determine the level of the user’s profile.
2. Set the TSCAUFLD field to the level of the user’s profile.
3. Check whether the user’s level matches the TSP level. As supplied, the level of the TSP is level1.

4. If the two levels do not match, process the ADDDATA, PROCESS, and USEREXIT lines to update the profile (in the shipped example, BYPASS COPYRIGHT? is set to YES) and also store the TSP level in the user profile. The last ADDDATA quits out of Information/Management to permanently save the updated profile.

5. If the two levels match, EXIT.

To modify this TSP:

1. Update the ADDDATA line with the changes you want to make to your users’ profiles.
2. Change level1 to level2.
3. Copy the updated TSP from a write panel data set to a read panel data set.

For more information on modifying TSPs, refer to the Panel Modification Facility Guide and the Terminal Simulator Guide and Reference.

Setting Values for the PDF Editor

If your users want to use the PDF editor for text processing, you can identify an invocation macro for them. When your user sets his user profile to use the PDF editor, it can automatically trigger Information/Management to run an invocation macro that you predetermine. You can use a single macro for all types of records and text, or you can specify a different macro for each type of text in each type of record. See “Specifying an Invocation Macro for the PDF Editor” on page 1-14 for more information.

You can also identify an ISPF panel for them to use with the PDF editor. This panel can display text type, record number, and other Information/Management data that you predetermine. You can use a single ISPF panel for all types of records and text, or you can specify a different panel for each type of text in each type of record. For information on changing the name of the ISPF panel or changing the ISPF variables that are set when using the PDF editor for freeform text, refer to “Changing Fields Displayed on Freeform Text Panels” in the checklists for complex modifications in the Panel Modification Facility Guide. For information on how to change the ISPF panel to display different ISPF variables, refer to the ISPF Dialog Developer’s Guide and Reference.

Note: This ISPF panel is used only with the PDF editor. It is ignored if the user uses the Information/Management editor or a workstation editor.

Store your ISPF panels in a data set that is in the ISPPLIB concatenation.

The default logical record length (LRECL) for the PDF editor is 132. You can customize this value by setting a variable before you call the invocation macro. See “Customizing the LRECL for PDF” on page 1-14 for more information.

No special considerations need to be made if your users want to use a workstation editor.
Setting Values for the PDF Editor

Specifying an Invocation Macro for the PDF Editor

The way Information/Management determines which, if any, invocation macro to use is by the setting of an ISPF variable called BLGITEXT. You can set this variable using program exit BLG01376. Refer to the Panel Modification Facility Guide for information on how this exit works.

To get you started, you can use PMF to look at panel BTN1S026. This panel calls program exit BLG01376 to set variable BLGITEXT to a value of BLGISMAC. BLGISMAC is the name of the invocation macro shipped with Information/Management.

After setting BLGITEXT, the panel calls the freeform text editor (program exit BLG01208).

Finally, the panel uses program exit BLG01376 to blank out the value of BLGITEXT. This ensures that this macro is not used for other record or text types.

You can call program exit BLG01376 anywhere in your panel flows, as long as it is on a control panel. If you want a different macro for each record type, you can call this program exit early in your record create or update panel flow and, optionally, blank it out at record file time.

If you want a different macro for each type of text in a record, you must insert a control panel (if one is not already there) in the panel flow after the user selects the text type. You must move the program exit from the selection panel to the new control panel. Again, panel BTN1S026 can give you an idea of how to set up the control line to call the editor from a control panel.

**Note:** The value of BLGITEXT is only used with the PDF editor. It is ignored if the user uses the Information/Management editor.

Store your macros in a data set that is listed in the SYSPROC DD statement concatenation.

Customizing the LRECL for PDF

Information/Management determines the LRECL for the temporary data set for the PDF editor from the setting of ISPF variable BLGIRECL. You can set this variable using program exit BLG01376 to set this variable before it calls the invocation macro for the PDF editor. Refer to the Panel Modification Facility Guide for information on how this exit works.

If BLGIRECL is not defined, not numeric, or not in the range of 1 through 244, an LRECL of 132 is used. If you assign BLGIRECL a value greater than 132, update the text display table panels to specify the same value so the text can be displayed in display mode. Also, if you assign BLGIRECL a value greater than 132, and you use an application running with an application program interface to retrieve records with the data set option for freeform text, only 132 characters of text per line is shown.

**Note:** The value of BLGIRECL is only used with the PDF editor. It is ignored if the user uses the Information/Management editor.
Preserving the Audit Trail for Freeform Text
The audit trail for freeform text cannot be preserved if an entire record is edited using the ISPF/PDF editor or a workstation editor. If you update the freeform text in an existing record using either the ISPF/PDF or a workstation editor, the text control data fields in the record are lost when you file it. Only the current control information is filed with the record.

As the program administrator, you can control the use of the ISPF/PDF editor and/or workstation editor by using the TEXTAUD keyword in the BLGPARMS session-parameters member macro. If TEXTAUD is specified as YES, panel BLG00105 appears when a user attempts to edit freeform text with the ISPF/PDF editor or workstation editor. This panel requires the user to select the type of edit (add or update) to perform on the text.

TEXTAUD also controls whether an API application can replace or delete existing freeform text. If TEXTAUD=YES is specified in the BLGPARMS session-parameters member macro, an application cannot replace or delete existing freeform text.
Defining Privilege Classes

This chapter describes privilege classes and suggests how to set up privilege classes to authorize users in your organization to perform certain tasks. It includes a series of scenarios that illustrate managing privilege classes.

Assigning Privilege Classes

A privilege class authorizes a user to perform specific tasks and access certain facilities. Your organization does not have to use privilege classes, but it is recommended that you create at least a master privilege class and one other class. The first privilege class that you create is given master authority. If you do not create a privilege class, users have authority to perform every task, including working with SRCs and privilege class records.

You can define a privilege class for either an individual or a group of users. Because it is often desirable for several users to have access to the same data (if they have similar or complementary job responsibilities), you can assign them to the same privilege class. For example, system operators and system programmers can share the same privilege class.

Optionally, you can authorize the use of Information/Management data through privilege classes. An Information/Management privilege class record identifies the privilege class to which a specific individual or group of individuals belongs. A user can perform only the functions permitted by that privilege class, such as accessing certain types of records in a database or making changes to certain types of records.

Each member of a privilege class is identified by an MVS logon identifier (user ID). Up to 24 user IDs can be associated with each privilege class; however, by using the panel modification facility (PMF), you can modify the panels to accept a larger number of user IDs. While it is possible for any user to log on to Information/Management, a user who is not a member of a privilege class is limited to displaying the overview, profile, and SRC data.

Categories of Users

Consider creating privilege classes that complement the task structure of your data processing organization. Privilege classes can identify the individuals or groups that represent:

- Functional units in the data processing organization
Categories of Users

- Change approvers
- Change reviewers
- Interested parties to a problem
- Information/Management administrators

The number and role of privilege classes depend on the size and structure of your organization. The privilege class for functional units can correspond directly to your data processing organization, or they can include selected groups and individuals. In either case, consider problem reporters, change requesters, coordinators, and assignees. Each group or individual that has change approval or reviewer responsibility should be identified in a privilege class.

Master Privilege Class

As a program administrator, you need universal authority. Therefore, you must be a member of the master privilege class. That is the first privilege class record you create in the Information/Management database. The master privilege class record should contain the user IDs of those individuals assigned as program administrators. (You can add or remove user IDs as required.) As a member of the master privilege class, you can use Information/Management’s full capabilities; you are automatically given all authorities for all record types, can use PMF to modify panels, and can perform certain error recovery functions.

You can also process any SRC regardless of whether the SRC lists the master privilege class as a class that can run the SRC. See Figure 2 on page 2-5 for lists of all the authorities.

You create the master privilege class record in the same way that you subsequently create other privilege classes. However, when you create the master privilege class, you cannot use the Transfer-to class field; you do not need to assign authority because you automatically have all authorities.

Note: The first privilege class that you enter into the database automatically gets all the authorities for Information/Management. You are not required to name that privilege class MASTER, but it is the “master” privilege class.

If you use PMF to create new authorities, the master privilege class gets them automatically.

Once a master privilege class exists, all users must be running under a privilege class to perform a function that is governed by authority. It is recommended that you create at least the master privilege class and one other class. Restrict the master privilege class to a few selected individuals, and assign the remaining users to other privilege classes.

Privilege Class Structure

The first consideration in creating privilege classes is the current or planned functional organization of your data processing installation. Because a privilege class can contain one or more user IDs and a user can be a member of one or more privilege classes, the key to deciding who belongs in what privilege class is to identify the tasks they need to perform.
The structure of your installation’s privilege classes should provide each user with the authority needed to perform the assigned system management activities. The two class structures are individual and group.

**Individual Privilege Classes**

Individual privilege classes identify users with unique responsibilities. For instance, a person responsible for resolving problems would be identified in a privilege class with the authority to create, update, assign, close, display, and delete problem records. The privilege class name could be identified in the **Assignee name** or **Transfer-to class** field in all problem records they would work with.

**Group Privilege Classes**

Group privilege classes identify users with shared responsibility. If, for instance, you have a group of people who manage problems, their user IDs should be identified in one privilege class that gives all of them the appropriate authorities for problem records.

At some organizations, problems are assigned to a department when they first occur. An owner is not specified at that time. Later, a member of the department selects the problem for solving and becomes the assignee. If you use this method, you can set up Information/Management in either of the following ways:

- At the time a record is created, the user enters the name of a privilege class in the **Assignee name** field. Later, a user who is a member of the assigned privilege class searches for all records assigned to the group. On finding a problem to solve, that user either:
  - Puts his or her name in the **Assignee name** field
  
  or

  - Assumes ownership by putting his or her individual privilege class name in the **Transfer-to class** field and filing the record. That user is then the only person (outside of those users defined in the master privilege class) who can update the record.

- To limit the number of users who can update the record, the user who creates the problem record names the appropriate group privilege class in the **Transfer-to class** field of the record.

Then, a member of the group uses the group’s privilege class to sign on to Information/Management and search for problems that are transferred to the group. When that user finds a problem to solve, he updates and transfers that problem to his individual privilege class by specifying it in the **Transfer-to class** field, and files the record. Until he again updates the record and files it using his individual privilege class, he, any other member of the group privilege class, and any member of the master privilege class can update the record. However, once he updates the record and files it using his individual privilege class, he becomes the owner and is the only person (outside of those users defined in the master privilege class) allowed to update the record.
Whenever you create a group privilege class, identify a group contact. The contact serves as the focal point for questions and issues about the privilege class’s responsibilities for particular records and program management tasks. The contact can also communicate information about a record or task to other members of the privilege class. Set up the selection criteria for contacts so that any user with the authority to create privilege classes can assign an appropriate contact.

The privilege class record contains fields to identify the contact’s name, phone number, and department. For a privilege class containing only one user ID, use these fields to supply information about that user.

**Example of Organization/Privilege Class Structure**

Figure 1 shows a privilege class structure for a data processing organization. In this organization, the Installation Controls department has the master privilege class. The Operations department uses group privilege classes, except for SSUPER, which is the system supervisor’s individual privilege class. The two programming departments each have a group class as well as two individual classes for members of the department. Table 4 on page 2-12 shows the authority assignments for this problem management organization.

![Diagram of privilege class structure](image-url)

**Figure 1. Privilege Class Structure for an Organization**
Understanding Authorities

Information/Management uses an authorization system to control the creation, update, display, and deletion of database records. The privilege classes you create are given varying amounts of authority. You can combine the authorities of these assigned privilege classes to:

- Perform a particular function for a given record type, such as create problem records
- Restrict the use of certain selections, input fields, or responses to input fields (such as the Assignee name field)
- Create and modify panels, dictionary display and update, and PMF reports

Figure 2 shows all the function- and field-level authorities for records and panels.

![Diagram of Privilege Classes and Authorities](image)

Some of the record categories shown in Figure 2 include more than one record type, such as:

- The change record category includes both change and activity records.
- The configuration record category includes hardware and software component and feature records and subcomponent records, as well as data center, system, and service organization records.
- The financial record category includes both hardware and software financial records.
Each remaining record category (problem, privilege class, SRC, and rules) involves only one record type.

When you create a privilege class, specify the types of authority it is to have. You can give the new privilege class the same or fewer authorities than the privilege class under which you are running, but you can never give it more.

**Function Level Authorities**

The function level authorities (entry, display, update, and delete) allow a user to perform specific functions.

**Display Authority**

To display, print, or report on any record type other than SRC records, a user must be running under a privilege class that contains display authority for that particular record type. However, a user can view any privilege classes that contain his own logon identifier when he selects Class on the Primary Options Menu.

**Entry Authority**

Entry authority is required to create a new database record. Entry authority also allows the user to copy a record. A privilege class cannot be authorized to create a particular record type unless it can also display that record type.

**Update Authority**

Update authority allows changes to a database record. A user must be running under a privilege class that has update authority for a given record type before he can change that type of record. A privilege class cannot be authorized to update a particular record type unless it can also display that record type.

This authority is also needed to transfer record ownership. For example, a user in the HELPDESK privilege class creates a record and transfers it to the SYSOPER privilege class. If the SYSOPER privilege class does not have update authority, the record cannot be transferred to that privilege class.

If a record is not owned, the user must be running under a privilege class that has update authority for that record type to update it.

If a record is owned, the user must be running under a privilege class that has update authority for that record type, and that privilege class must own the record or be the transfer-to privilege class before the user can update the record. No other privilege class (except the MASTER privilege class) can update an owned record. See “Deciding Whether to Use Record Ownership” on page 2-13 for an explanation of record ownership.

**Delete Authority**

Delete authority is required to delete a record from the database. A privilege class cannot be authorized to delete a particular record type unless it can also display that record type. A user cannot delete a record
if any other records reference that record. For example, one of the fields in a problem record is **Original problem number**. If record number 3 in your database has 2 in that field, it might mean that problem 2 was not fixed correctly, and problem 3 resulted from the incorrect fix. You cannot delete problem record 2, because problem record 3 makes a reference to it.

When you delete a parent record of a parent/child relationship, the child records are also deleted.

If a record is not owned, the user must be running under a privilege class that has delete authority for that record type in order to delete the record.

If a record is owned, all of the following conditions must exist before the user can delete it:

- The user must be running under a privilege class that has delete authority for that record type.
- The privilege class in which the user is running must own the record.
- The record cannot be referenced by any other record.

No other privilege class (except the MASTER privilege class) can delete that record.

**Field Level Authorities**

The assignment and close authorities for problem, change, and activity records designate certain fields that can be completed only by users with the appropriate authority. Using PMF, you can modify these authorities or include additional authorities for other fields or selections. Field level authority is controlled through authorization codes for fields and privilege classes that permit their usage.

**Assignment Authority**

Assignment authority is required to enter assignment information for problem or change records. A privilege class cannot be given assignment authority unless it has display authority and either entry or update authority for the record type.

A user must have assignment authority for a given record type before any assignment data can be entered in the fields listed in Table 1 on page 2-8.
Understanding Authorities

Even though a privilege class can own and have entry or update authority for a particular record, its members can enter assignment data only if the privilege class has assignment authority.

Close Authority

Close authority is required to enter close information for problem or change records. A privilege class cannot be given close authority unless it has display authority, and either entry or update authority for the record type.

A user must have close authority for a given record type before any close data can be entered in the fields shown in Table 2.

Table 1. Problem and Change Record Fields Requiring Assignment Authority

<table>
<thead>
<tr>
<th>Problem Records</th>
<th>Change Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignee name</td>
<td>Assignee name</td>
</tr>
<tr>
<td>Assignee department</td>
<td>Assignee department</td>
</tr>
<tr>
<td>Assignee phone</td>
<td>Assignee phone</td>
</tr>
<tr>
<td>Date assigned</td>
<td>Date assigned</td>
</tr>
<tr>
<td>Time assigned</td>
<td>Time assigned</td>
</tr>
<tr>
<td>Assignment status</td>
<td></td>
</tr>
<tr>
<td>Assignment number</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Problem and Change Record Fields Requiring Close Authority

<table>
<thead>
<tr>
<th>Problem Records</th>
<th>Change Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolved by</td>
<td>Closed by</td>
</tr>
<tr>
<td>Resolver department</td>
<td>Closer department</td>
</tr>
<tr>
<td>Resolver phone</td>
<td>Closer phone</td>
</tr>
<tr>
<td>Resolver class</td>
<td>Closer class</td>
</tr>
<tr>
<td>Date closed</td>
<td>Completion date</td>
</tr>
<tr>
<td>Time closed</td>
<td>Completion time</td>
</tr>
<tr>
<td>Cause code</td>
<td>Completion code</td>
</tr>
</tbody>
</table>

Even though a privilege class can own and have entry or update authority for a particular record, its members can enter close data only if the privilege class has close authority.

PMF Authorities

Panel modification requires PMF authority. A user must be running under a privilege class that has the associated PMF authority to create, update, copy, or delete panels; to update or display the dictionary data set; to print PMF reports; or to list panel data sets.

The PMF authorities are selection-level authorities; they correspond to the options on the PMF entry panel.
It is recommended that you restrict PMF authorities to individuals who require them. Keep in mind, however, that in order to use PMF efficiently, a user needs all authorities.

**Update Authority**
Panel update authority is required to create or update panels. With panel update authority, a user can make external or internal panel modifications and use the **Panel set** selection to create a new set of panels.

**Panel Copy Authority**
Panel copy authority allows a user to duplicate a panel on the same or a different panel data set. (If you duplicate a panel on the same data set, you must rename it.) To modify a copied panel, the user must be running under a privilege class that has panel update authority.

Copy authority is required to move updated panels from the write panel data set to a read panel data set to make them available for use.

**Panel Delete Authority**
Panel delete authority is required to delete a panel from a panel data set.

**Dictionary Display Authority**
Dictionary display authority allows a user to display prefix and s-word information in the dictionary data set. With this authority, more than one user can display the dictionary, even while it is being updated.

**Dictionary Update Authority**
Dictionary update authority allows a user to display, update, add, and delete prefix and s-word information in the dictionary data set. A maximum of two users can update the dictionary at the same time: one user is allowed to update s-word information and the other user is allowed to update prefix information.

**PMF Reports Authority**
PMF reports authority allows a user to run PMF reports. The PMF reports provide panel and dictionary information that the user needs to plan panel modifications.

**Panel List Authority**
Panel list authority allows a user to display the list of panel data set names and data set definition labels referenced by the current session-parameters member. With panel list authority, a user can look at the panels contained in the panel data set and perform any of that user’s authorized PMF functions on the panels listed. In order to have panel list authority, a privilege class record must also have all the other PMF authorities.
**Database Administration Authority**

Database administration (DBADMIN) is an Information/Management System application authority for the Information/Management program administrator. Having DBADMIN authority enables the program administrator to use the following functions in Information/Management:

**ALIAS Record**

The ALIAS record is used to define alias names for TSPs and TSXs. You must have DBADMIN authority to create, change, or delete this record, but you do not need this authority to display or print the ALIAS record.

**CHECK IN**

The CHECK IN function finds records that are marked *checked out* by an application running with the application program interface and checks the records back in. Use this function only if the application ended without checking the records in.

**COMMAND Record**

The COMMAND record is used to define aliases for commands and to restrict usage of commands. You must have DBADMIN authority to create, change, or delete this record, but you do not need this authority to display or print the COMMAND record.

**Data Attribute Records**

Data attribute records are used to define and store your Information/Management data model using records rather than panels. You must have DBADMIN authority to create, update, and delete these records.

**Data View Records**

Data view records are used to define and store your Information/Management data model using records rather than panels. You must have DBADMIN authority to create, update, and delete these records.

**DBCLEANUP**

DBCLEANUP is a database administration function that cognizes and files any records that are marked busy or deleted as a result of a system outage or ABEND.

**LOGSAVE Record**

The LOGSAVE record is used by the Automatic Log Save Facility, which enables you to propagate Information/Management data to another Information/Management database. You must have DBADMIN authority to create, change, or delete a LOGSAVE record, but you do not need this authority to display and print a LOGSAVE record.

**Validation Records**

Validation records are used in conjunction with assisted-entry panels when you want to present users with a list of valid values to choose from. You must have DBADMIN authority to work with validation.
Defining Privilege Classes

Understanding Authorities

records; you must have PMF authority to change the panel flow to access the validation record.

**Universal Partition Access Authority**

Data in an Information/Management database can be organized into logical partitions within a single physical database. A privilege class authority called **Universal Partition Access** provides access to all logical partitions. Any user with this authority has access to all records in any partition in the Information/Management database. For more information on logical databases, see “Logical Database Partitioning” on page 4-1.

**Assigning Authorities**

When you assign authorities, consider the tasks that each of your users perform. If you organized your privilege classes according to this consideration, it is easy to assign the necessary authority to each class. Keep in mind that the fewer users with a particular authority, the more control you have over your data. This is particularly important with delete authority. If too many privilege classes can delete records, there is a greater chance that you could lose valuable data.

To allow a user to perform a specific task, you can give him a combination of authorities. For example, when a user updates a problem record, a summary of the problem is first displayed so the user can verify that it is the correct record. Thus, for the user to update this problem, he must first be allowed to display it. In this case, you would give the user authority to display and to update problem records.

Table 3 summarizes the authority levels required for specific tasks.

<table>
<thead>
<tr>
<th>If you want to:</th>
<th>Display</th>
<th>Entry</th>
<th>Update</th>
<th>Assignment</th>
<th>Close</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display records</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Create new records</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Update existing records</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assign records when created</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Assign records when updated</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Close records when created</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Close records when created</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete records not owned</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Delete records owned</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

With only display authority, the user can only display records. However, for other tasks, the user must have authority levels in appropriate combinations, always including display authority.
Example of Authority Assignments for Problem Management Users

The matrix in Table 4 shows the authority assignments for the problem management organization shown in Figure 1 on page 2-4.

<table>
<thead>
<tr>
<th>CLASS</th>
<th>PROBLEM</th>
<th>SRC</th>
<th>PRIVILEGE CLASS</th>
<th>PMF</th>
<th>DB</th>
<th>UNIV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DIS</td>
<td>EN</td>
<td>UP</td>
<td>AS</td>
<td>CL</td>
<td>DE</td>
</tr>
<tr>
<td>MASTER</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>HELPDESK</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>TAPEOPER</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISKOPER</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSOPER</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSUPER</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>PRODCNTL</td>
<td>Y</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYSPROG</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>SPROG1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>SPROG2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>APPLPROG</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>APROG1</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>APROG2</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td></td>
</tr>
</tbody>
</table>

DIS=DISPLAY  EN=ENTRY  UP=UPDATE  AS=ASSIGNMENT  CL=CLOSE

Several factors are worth noting about the assignments in Table 4.

- The master privilege class has all authorities. It is the only class having authority to delete problem records, thus ensuring that valuable data is not deleted and is available for reports.
- SSUPER has all authorities except problem delete. SSUPER represents his group and is responsible for creating and maintaining SRCs, privilege classes, and panels for his group.
- SPROG1 and APROG1 have all authorities except problem delete and PMF. They are also representatives for their groups. They have the same responsibilities as does SSUPER, but they cannot modify panels.
- HELPDESK closes problems that they resolve on the spot.
- Other groups have varying levels of authority.
- All classes are authorized to display privilege classes to stay informed of authority changes.
Deciding Whether to Use Record Ownership

Your installation can decide to control changes to data records through record ownership. The decision to use ownership for records is based on the amount of data security you want to have. If it is important to you to limit the number of users who can update or delete a particular record, that record should be transferred to an owner. However, because Information/Management neither restricts nor enforces the use of record ownership, it is your responsibility to establish your own standards and enforce those standards within your installation.

If you decide to use ownership, each privilege class to which a record can be transferred must be given update authority for that type of record so that the record can be transferred to it. Once ownership is established, only users defined in the owning, the transfer-to, or the master privilege classes are authorized to update the record. Likewise, if a record is owned, only the master privilege class or the owning privilege class (if it has delete authority for that type of record) can delete it.

To establish initial ownership of a record, a member of a privilege class with create or update authority for the record type must enter the name of a privilege class in the record’s Transfer-to class field and file the record. (Note that the record can be transferred only to a privilege class that has update authority for that type of record.) When the record is filed, the privilege class that created the record (if it is a new record) or updated the record is assigned ownership immediately. If a user transfers the record to his current class, the Transfer-to class field is cleared when the record is filed. If a user transfers the record to a class other than his current class, the Transfer-to class field is cleared and ownership of the record is reassigned when a member of the transfer-to privilege class accesses the record in update mode, makes a change to the record, and subsequently files it.

If the user who creates a record does not specify an owner, any user with update authority for that record type can assign ownership of the record to himself at any time. Once ownership is assigned, only a user identified in the owning privilege class or the master privilege class can transfer ownership to another privilege class.

An Owning Partition Name can be added to each record so that records can be grouped into logical partitions. Information about this aspect of record ownership can be found in “Logical Database Partitioning” on page 4-1.

Managing Privilege Classes

The basic tasks performed in managing privilege classes are:

- Defining individual and group privilege classes
- Determining all privilege class names
- Displaying authorities for a privilege class
- Deleting members from and adding members to a privilege class
- Deleting a privilege class

To learn how to perform these tasks, go through the step-by-step scenarios on the following pages.
Defining an Individual Privilege Class

In this scenario, you define a privilege class for Tim Bennett, a user, who is the Problem/Change Coordinator. To fulfill his job responsibilities, the Problem/Change Coordinator requires all authorities for problems, changes, privilege classes, and SRCs. Tim also performs some program administration duties when the program administrator is absent. Therefore, he needs database administration authority. (If database partitioning is used, Tim could also be given universal authority, but only if he needs access to all of the records in the Information/Management database. See “Logical Database Partitioning” on page 4-1 for details.)

**Note:** You can only create a privilege class having a subset of the authorities defined in the privilege class in which you are currently running.

If the application specified in the upper-right corner of the screen is Management, change the application setting to System. Type:  
3,1
and press Enter.

From BLGOEN10, Primary Options Menu, select **Entry**. (Notice that you do not use the **Class** selection to create or to search privilege class records.) Type 5 and press Enter.

```
BLGOEN10   --- PRIMARY OPTIONS MENU ---   APPLICATION: SYSTEM
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. DBADMIN........Perform database administration.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997

=> 5
```
Select **Class**. Type 1 and press Enter.
On BLG0J100, the Class Description Entry panel, complete the required fields and any others. In the **User ID** field, put the MVS logon identifier for Tim Bennett. In this example, that is SYSPRG1. After typing the data in the appropriate fields, press Enter.

(The **Primary partition id** field is used for partitioning your database. See “Logical Database Partitioning” on page 4-1 for details.)

To leave the panel, type **end** and press Enter.
The Class Summary panel gives a summary in the upper portion and lists all categories of data that can be entered in the lower portion. This panel is redisplayed to allow you to add as much data as you want until you are ready to file the record.

Now to specify the authority, select **Class authority**. Type 2 and press Enter.
Assign authority for system application records. Select System. Type 1 and press Enter.

You select System because SRCs and privilege classes are system application records, and because database administration is a System application authority.

Select Class. Type 1 and press Enter.
From this panel, complete the necessary fields by typing yes for all the functions, because Tim Bennett requires all authorities in order to perform his job.

To exit, type END and press Enter.
From this panel, type yes in the field for each function you want Tim Bennett to be able to do.

To leave this panel type END and press Enter.

```
BLGOJ290 SRC AUTHORITY CLASS: PCHCOORD

Enter 'YES' to permit this privilege class to perform the specified function.

1. SRC entry................ yes
2. SRC update............... yes
3. SRC delete............... yes

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

>>> end
```

You return to BLGOJ212, the Authority Entry panel.

Select DBADMIN. Type 3 and press Enter.

```
+ BLGOJ212 -------------- AUTHORITY ENTRY ----------------- 1 OF 1-+

SELECT THE TYPE OF AUTHORITY TO BE ENTERED

OPTIONS:

1. CLASS......Authority for privilege class records.
2. SRC........Authority for SRC (stored response chains.)
3. DBADMIN....Authority for database administration.
4. UNIVERSAL..Authority to access all logical partitions

+- Enter your choice, or type ;END or ;CANCEL to leave this panel.+-

>>> 3
```
Now, you are ready to give database administration authority.
Type yes and press Enter.

```
/+ BLG6AUDB --- DATABASE ADMINISTRATION AUTHORITY LEVEL ---- AUTH/-+
USE....Reply YES to indicate authority for this class.
NOTE...Reply ¬ to remove database administration authority for
this class. Reply ;BA P to return to the options
panel without modifying the authority.
REPLY: YES  - The privilege class is allowed to perform
database administration functions.
¬  - Remove database administration authority
from this class.
+----------------------- REPLY AS DEFINED -------------------------+
```

```===>
yes
``` You return to BLG0J212, the Authority Entry panel.
Type end and press Enter because you are finished giving System application authority.

```
/+ BLG0J212 ----------- AUTHORITY ENTRY ------------- 1 OF 1-+
SELECT THE TYPE OF AUTHORITY TO BE ENTERED
OPTIONS: 
1. CLASS......Authority for privilege class records.
2. SRC........Authority for SRC (stored response chains.)
3. DBADMIN....Authority for database administration.
4. UNIVERSAL..Authority to access all logical partitions
+- Enter your choice, or type ;END or ;CANCEL to leave this panel.-+
```

```===>
end
```
Now, you are ready to give Management application authorities.
Select **Management**. Type 2 and press Enter.

```plaintext
+ BLGGH211 -------------- AUTHORITY ENTRY ------------------- 1 OF 1-+

SELECT THE TYPE OF AUTHORITY TO BE ENTERED

OPTIONS:
1. SYSTEM........Authority for System application.
2. MANAGEMENT.....Authority for Management application.

+- Enter your choice, or type ;END or ;CANCEL to leave this panel.---+

###> 2
```

Select **Problem** to give authority for problem records. Type 1 and press Enter.

```plaintext
+ BLGGH210 -------------- AUTHORITY ENTRY ------------------- 1 OF 1-+

SELECT THE TYPE OF AUTHORITY TO BE ENTERED

OPTIONS:
1. PROBLEM........Authority for problem records.
2. CHANGE..........Authority for change records.
3. CONFIG..........Authority for configuration records.
4. FINANCIAL......Authority for financial records.
5. PMF.............Authority for panel maintenance.
6. RULES..........Authority for escalation rules records.

+- Enter your choice, or type ;END or ;CANCEL to leave this panel.---+

###> 1
```
Type yes for as many functions as you want Tim Bennett to be able to do. Because he is the problem coordinator, he needs all authorities.

To exit, type end and press Enter.

```
BLG0J200 PROBLEM AUTHORITY CLASS: PCHCOORD
Enter 'YES' to permit this privilege class to perform the specified function.

1. Problem entry............. yes
2. Problem update............ yes
3. Problem delete............ yes
4. Problem display........... yes
5. Problem assignment data... yes
6. Problem close data........ yes

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

```
Managing Privilege Classes

From this panel, type yes for all the functions, because Tim Bennett requires all authorities in order to perform his job as change coordinator.

To leave this panel and return to the Class Summary panel, type end,end,end and press Enter.

---

Up to this point, you have described a privilege class and given it five kinds of authority. Now, provide some descriptive text about this class.

Select Freeform text and notes. Type 8 and press Enter.
Defining Privilege Classes

Managing Privilege Classes

Select **Description**. Type 1 and press Enter.

+---------- PRIVILEGE CLASS TEXT ENTRY ----------+ 1 OF 1 +
PRIVILEGE CLASS TEXT DIALOG ENTERED, SELECT OPTION
OPTIONS:
1. DESCRIPTION....Description of this privilege class.
2. NOTES..........Miscellaneous notes for privilege class.
3. ADDRESS........Address of contact name or department.
4. END............Freeform text entry dialog is complete.

If your profile setting for **Editor selection** is INFO, the following panel appears. For more information on selecting an editor, refer to the *User’s Guide*.

Type the text on the full-screen text editor panel.

When the text is complete, save it and bypass the text selection panel. Type `end,,` and press Enter.
Once again, the Class Summary panel appears. If you want to include other users in this class, you can select *Eligible users* from this panel and add them. (If your database is logically partitioned, you can use selections 4 and 5 to identify partition information for this privilege class.) For this example, save the data you have created. Select *File record*. Type 9 and press Enter.

```
User ID 1.............. SYSPRG1_ Owning priv. class.... ________
Transfer-to class..... Entry priv. class..... ________
Contact name.......... BENNETT Date entered......... ________
Contact phone......... 555-2493 Time entered......... ________
Contact dept.......... ___________ Date last altered.... ________
Location code.......... ________ Primary partition..... _________
Description............ PROBLEM AND CHANGE COORDINATOR CLASS
```

Select one of the following, type END to save your changes, or type CANCEL to discard your changes.
3. Eligible users.
4. Secondary partition list.
5. Sharing partition list.

```
==> 9
```

Write down the record ID (name) assigned to the record. The record ID is the privilege class name that you entered on panel BLG0J100, Class Description Entry. The record ID for this privilege class is PCHCOORD.
Defining a Group Privilege Class

As the Problem/Change Coordinator, Tim Bennett, needs to give three people at the user help desk authority to open, update, assign, and close problems.

If the application specified in the upper-right corner of the screen is Management, change the setting to System. Using an immediate response chain (IRC), type 3,1 and press Enter.

Select Entry from the Primary Options Menu and Class from the System Record Entry panel. Type 5,1 and press Enter.

```
BLGOEN10 --- PRIMARY OPTIONS MENU --- APPLICATION: SYSTEM

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. DBADMIN........Perform database administration.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997

***> 5,1
```
Complete the data on the Class Description Entry panel.

Specify the user ID of one user.

Transfer the record to your privilege class to be the owner (you are the only person besides the master privilege class who can update or delete this record).

To leave the panel, type end and press Enter.
To add two more users, select **Eligible users**. Type 3 and press Enter.

Type an IRC for the second and third user IDs, and two commas as shown:

2,hdjanet,3,hdmarge,,

The IRC enters the information and returns you to the Class Summary panel. Press Enter.

---

```
To add two more users, select **Eligible users**. Type 3 and press Enter.

Type an IRC for the second and third user IDs, and two commas as shown:

2,hdjanet,3,hdmarge,,

The IRC enters the information and returns you to the Class Summary panel. Press Enter.
```
Managing Privilege Classes

To give this privilege class its authority, select **Class authority**. Type 2 and press Enter.

```
+ BLG0J211 -------------- AUTHORITY ENTRY --------------- 1 OF 1+

SELECT THE TYPE OF AUTHORITY TO BE ENTERED

OPTIONS:

1. SYSTEM........Authority for System application.
2. MANAGEMENT.....Authority for Management application.

+- Enter your choice, or type ;END or ;CANCEL to leave this panel.+-

====> 2
```

Select **Management**. Type 2 and press Enter.
Select **Problem** to give authority for problem records. Type 1 and press Enter.

Type YES for problem entry, update, display, close, and assignment authorities.

**Note:** Display authority is required for the other authorities.

To leave the panel and return to the Class Summary panel, type `end,end,end` and press Enter.
To provide descriptive text for the user help desk personnel’s privilege class, select **Freeform text and notes**. Type 8 and press Enter.

\[
\begin{array}{|lr|}
\hline
\text{BLG0JU00} & \text{CLASS SUMMARY} & \text{CLASS: HELPDESK} \\
\hline
\text{User ID 1............... HDMARK} & \text{Owning priv. class....} & \\
\text{Transfer-to class....... PCHCOORD} & \text{Entry priv. class....} & \\
\text{Contact name.......... BENNETT} & \text{Date entered........} & \\
\text{Contact phone......... 555-2493} & \text{Time entered........} & \\
\text{Contact dept..........} & \text{Date last altered....} & \\
\text{Location code.........} & \text{Primary partition.....} & \\
\text{Description.......... HELP DESK PRIVILEGE CLASS} & & \\
\hline
\end{array}
\]

Select one of the following, type END to save your changes, or type CANCEL to discard your changes.

1. Class description.
2. Class authority.
3. Eligible users.
4. Secondary partition list.
5. Sharing partition list.
8. Freeform text and notes.

\[\Rightarrow 8\]

Select **Description**. Type 1 and press Enter.

\[
\begin{array}{|lr|}
\hline
\text{BLG0J010} & \text{PRIVILEGE CLASS TEXT ENTRY ---------1 OF 1----------} +
\hline
\text{PRIVILEGE CLASS TEXT DIALOG ENTERED, SELECT OPTION} &
\hline
\text{OPTIONS:} & \\
1. DESCRIPTION...Description of this privilege class. & \\
2. NOTES..........Miscellaneous notes for privilege class. & \\
3. ADDRESS........Address of contact name or department. & \\
4. END............Freeform text entry dialog is complete. & \\
\hline
\end{array}
\]

\[\Rightarrow 1\]
If your profile setting for Editor selection is INFO, the following panel appears.
Type the description text on the full-screen text panel.
Save the text and return to the Class Summary panel. Type end,, and press Enter.

To save the privilege class record, select File record. Type 9 and press Enter.
Write down the privilege class name you assigned for future use. The name of the privilege class in this example is HELPDESK.
Determining All Privilege Class Names

Mark, Janet, and Margaret at the user help desk require an updated report of all privilege classes and the users assigned to them. You perform a search to print the required report.

From the Primary Options Menu for the System application, select Inquiry. Type 6 and press Enter.
Select **Class**. Type 1 and press Enter.
This panel appears if you are using the structured search method of searching. If you are using the quick search method of searching, panel BLG0K190 appears. Refer to the User's Guide for information about methods of searching.

You want all privilege classes included; therefore, you need not further qualify the search argument.

To get the report, issue the REPORT command. Type report and press Enter.
The Page Summary report provides the data needed by the user help desk. Because the report you want is common to all record types, you want a report in the general category.

Select **General.** Type 1 and press Enter.

```
BLGW500 REPORT ENTRY
Identify the type of Info/Management report to be created.
1. GENERAL.........Summary reports for all applications.
2. PROBLEM.........Problem management reports.
3. CHANGE..........Change management reports.
4. CONFIG..........Configuration management reports.

8. USER RFT........Specify user report format table name.
10. BROWSE/PRINT....Browse or print existing report data set.

SELECT ITEM

BLG15001I The REPORT command is using database 5.
==> 1
```
Now, select **Page Summary**. Type 3 and press Enter.

If your user profile does not have output destination settings already set up, the next panel you see asks you to enter output destination information. For this example, assume you already have output destination information set up in your profile. Refer to the *User's Guide* for more information on output destinations.

You return to panel BLG0K000 where a message tells you that your report was successfully written.
Displaying Authorities for a Privilege Class

In this scenario, you need to know what authorities the help desk group has. However, you do not remember the name of the privilege class. First, perform a search to identify the class and then display the class.

From the Primary Options Menu for the System application, perform a search for privilege class records. Type 6,1,search and press Enter.

```
BLGOEN10 --- PRIMARY OPTIONS MENU --- APPLICATION: SYSTEM
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. DBADMIN........Perform database administration.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997

==> 6,1,search
```
Managing Privilege Classes

The first page of the search results list contains the class you want, record 3.
Select the class for display. In the line command area for record 3, type s and press Enter.

Select Authority display. Type 1 and press Enter.
Because you want to see all of the authorities this record has, select **Summary**. Type 3 and press Enter.

![authority display dialog]

```plaintext
+- BLG0Q211 -------------- AUTHORITY DISPLAY -------------- 1 OF 1 -+

SELECT THE AUTHORITY DISPLAY DIALOG TO BE ENTERED

OPTIONS:

1. SYSTEM........Authority for System records.
2. MANAGEMENT.....Authority for Management records.
3. SUMMARY........Summary of all assigned authority.

+- Enter your choice, or type ;END or ;CANCEL to leave this panel.---+

```}

Now a summary of all authorities appears. To return to the search results list, type `end,end,end` and press Enter.

![authority summary]

```
BLG0Q213

<table>
<thead>
<tr>
<th>PROBLEM CHANGE</th>
<th>RECORD CREATE</th>
<th>RECORD UPDATE</th>
<th>RECORD DELETE</th>
<th>RECORD DISPLAY</th>
<th>RECORD ASSIGN</th>
<th>RECORD CLOSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CONFIG</th>
<th>FINANCIAL</th>
<th>CLASS</th>
<th>SRC</th>
<th>RULES</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PANEL UPDATE</th>
<th>DICTIONARY DISPLAY</th>
<th>DICTIONARY UPDATE</th>
<th>PANEL COPY</th>
<th>PANEL DELETE</th>
<th>PMF REPORTS</th>
<th>PANEL LIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATABASE ADMINISTRATION ACCESS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

To return to Authority Entry Display, type END or CANCEL.

```
Deleting One Member of a Privilege Class

In this scenario, three employees are members of the HELPDESK privilege class. One of them, Margaret, no longer requires access to Information/Management. Margaret’s user ID is HDMARGE.

You can start from the Primary Options Menu for the Management application and type the UPDATE command with the privilege class identifier UPD R HELPDESK and press Enter.
To modify eligible user information, select **Eligible users**. Type 3 and press Enter.

<table>
<thead>
<tr>
<th>BLG0JU00</th>
<th>CLASS SUMMARY</th>
<th>CLASS: HELPDESK</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID 1........... HDMARK</td>
<td>Owning priv. class....</td>
<td></td>
</tr>
<tr>
<td>Transfer-to class..... PCHCOORD</td>
<td>Entry priv. class..... PCHCOORD</td>
<td></td>
</tr>
<tr>
<td>Contact name........ BENNETT</td>
<td>Date entered........ 04/28/93</td>
<td></td>
</tr>
<tr>
<td>Contact phone......... 555-2493</td>
<td>Time entered........ 11:00</td>
<td></td>
</tr>
<tr>
<td>Contact dept..........</td>
<td>Date last altered..... 04/28/93</td>
<td></td>
</tr>
<tr>
<td>Location code.........</td>
<td>Primary partition.....</td>
<td></td>
</tr>
</tbody>
</table>

Description........ HELP DESK PRIVILEGE CLASS

Select one of the following, type END to save your changes, or type CANCEL to discard your changes.

3. Eligible users.
4. Secondary partition list.
5. Sharing partition list.

```===> 3
```

To delete Margaret’s user ID, type 3,, and press Enter.

To leave the panel, type END and press Enter.

<table>
<thead>
<tr>
<th>BLG0J300</th>
<th>CLASS USER ENTRY</th>
<th>CLASS: HELPDESK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter eligible user IDs; cursor placement or input line entry allowed.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
1. User ID 1........<R> HDMARK
2. User ID 2........... HDJANET-
3. User ID 3........... HDMARGE-
4. User ID 4............
5. User ID 5............
6. User ID 6............
7. User ID 7............
8. User ID 8............
9. User ID 9............
10. User ID 10.........
11. User ID 11.........
12. User ID 12.........
13. User ID 13.........
14. User ID 14.........
15. User ID 15.........
16. User ID 16.........
17. User ID 17.........
18. User ID 18.........
19. User ID 19.........
20. User ID 20.........
21. User ID 21.........
22. User ID 22.........
23. User ID 23.........
24. User ID 24.........
```

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

```===> 3,```

2-43
To save the updated privilege class record, select **File record**.
Type 9 and press Enter.

```
<table>
<thead>
<tr>
<th>BLG0JU00</th>
<th>CLASS SUMMARY</th>
<th>CLASS: HELPDESK</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID 1.............. HDMARK</td>
<td>Owning priv. class....</td>
<td></td>
</tr>
<tr>
<td>Transfer-to class....... PCHCOORD</td>
<td>Entry priv. class..... PCHCOORD</td>
<td></td>
</tr>
<tr>
<td>Contact name............ BENNETT</td>
<td>Date entered.......... 04/28/93</td>
<td></td>
</tr>
<tr>
<td>Contact phone........... 555-2493</td>
<td>Time entered.......... 11:00</td>
<td></td>
</tr>
<tr>
<td>Contact dept...........</td>
<td>Date last altered..... 04/28/93</td>
<td></td>
</tr>
<tr>
<td>Location code...........</td>
<td>Primary partition.....</td>
<td></td>
</tr>
<tr>
<td>Description............ HELP DESK PRIVILEGE CLASS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Select one of the following, type END to save your changes, or type CANCEL to discard your changes.
1. Class description.
2. Class authority.
3. Eligible users.
4. Secondary partition list.
5. Sharing partition list.
6. Class authority.
7. Eligible users.
8. Freeform text and notes.

```===> 9 ==```

### Deleting a Privilege Class

In this scenario, your company is reorganizing its operations and wants to eliminate the Help Desk department.

You have delete authority for the HELPDESK privilege class. All records that are owned by privilege class ID HELPDESK have been transferred or deleted. You know the privilege class record identifier, and you are currently using the Management application, so you begin at the Primary Options Menu after issuing the INITIALIZE command. It is not necessary that you start from this panel. You can delete by command from any panel.

From the Primary Options Menu, issue the DELETE command with the record ID.
Type:
```
delete r helpdesk
```
and press Enter.
Verify the delete. Select **Verify delete request**. Type 2 and press Enter.

A message tells you the record is deleted.

---

```plaintext
>>> delete r helpdesk
```

---

```plaintext
User ID 1.............. HDMARK
Transfer-to class...<H> PCHCOORD
Contact name......... BENNETT
Contact phone......... 555-2493
Contact dept......... ___________
Location code........ ________
Description............ HELP DESK PRIVILEGE CLASS

Select one of the following, type END to save your changes, or type CANCEL to discard your changes.

1. Cancel delete request.
2. Verify delete request.

```
Understanding Your Information/Management Database

This chapter provides an overview of the Information/Management database and describes how you can take advantage of the various options provided by Information/Management to set up a database that best suits your organization’s needs. In addition, it discusses the methods available within Information/Management to help you back up and recover your database.

The Databases

Users of Information/Management can be interactive users, batch jobs, or applications that call API transactions. As a user proceeds through a set of panels, making selections and entering data, Information/Management collects the structured words (s-words) and prefix words (p-words) that correspond to the fields and data being entered. When the user files the record, the s-word and p-word data used for searching goes into the SDIDS. That data, along with everything else, goes into the SDDS.

These data sets form the actual database for Information/Management. The records created by users are stored in the structured description data set (SDDS) in the form of structured descriptions. The records are located by keywords stored in the structured description index data set (SDIDS).

The structured description log data set (SDLDS) is an optional data set used as a backup for records written to the SDDS.

The pairing of an SDDS and SDIDS constitute a database. Information/Management uses two types of databases: Information/Management format databases and user-defined format databases. Both types of formats are made up of SDDS and SDIDS data sets. Databases 0, 1, 2, and 3 are user-defined format databases; databases 4, 5, 7, 8, and 9 are Information/Management format databases. Database 5 is a read/write database. The other databases are read-only databases.

The Information/Management read/write database can also contain an SDLDS. The two formats differ in the way that you enter data into the database. You enter data into an Information/Management format read/write database interactively through Information/Management panels or dialogs or through an API. A user-defined database receives data loaded through a utility program (BLGOZUD). The BLGOZUD utility enables you to load data into the database, which you can search, display or print, but you cannot update the records in a user-defined database. A detailed
Structured Description Data Set (SDDS)
The Structured Description Data Set (SDDS) is made up of from 1 to 100 VSAM key-sequenced data sets (clusters) that contain the data records. For an Information/Management format database, the SDDS contains the information supplied by the user (either interactively through panels or using API programs) or collected by Information/Management. Each Information/Management SDDS logical record 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.

Information/Management requires that you define an SDDS. The SDDS of an Information/Management format database can consist of from 1 to 100 clusters. A 100-cluster SDDS enables users to store up to 400 GB (GB equals 1,073,741,824 bytes) of information in a database.

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

Figure 3 on page 3-3 illustrates the way logical records are filed in multiple data sets, using five VSAM data sets as an example, to help you understand how records are stored in multiple data sets.

Structured Description Index Data Set (SDIDS)
The Structured Description Index Data Set (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 logical record in the SDIDS contains a searchable word and position markers to indicate which records in the SDDS contain that word. Each logical record can consist of one or more physical VSAM records. This is true for both Information/Management 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. The SDIDS data is compressed.

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

Figure 4 on page 3-4 shows the relationship between the words in the SDIDS and the records in the SDDS.

Structured Description Log Data Set (SDLDS)
The Structured Description Log Data Set (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.
Information/Management Database

The Databases

Figure 3. Logical Record Locations in an SDDS.

A logical record in Information/Management consists of one or more physical VSAM records. Each physical VSAM record that is related to the same Information/Management logical record has the same SDDS position number (also called a root VSAM key or VSAM sequence number). The SDDS position number is the second through fourth byte of the physical VSAM key. All physical VSAM records with the same SDDS position number are stored in the same VSAM data set.

The SDLDS is a VSAM relative record data set that consists of a single VSAM cluster and contains fixed-length records. Only the Information/Management read/write database (database 5) has an SDLDS.
This figure illustrates, for example, that if you enter a search argument of WORD1, the search mechanism reads the SDIDS record for WORD1 and checks the bit string to determine which records contain WORD1. If you enter a search for WORD1 or WORD3 (to retrieve records with both words), the search mechanism reads the SDIDS for WORD1 and WORD3 and performs a logical AND operation on the bit string of each word.

Your Database Options

One of the advantages of using the Information/Management database is its enormous storage capacity. You can store a significant amount of data for tracking and reporting at an enterprise level. Another advantage is its ability to be customized to suit your specific needs and business processes. Not all organizations have the same needs for database access, searching, storage, or performance. If you are new to Information/Management, or if you are an existing user looking at ways to improve your database operation, you should consider the options described in this section when evaluating what choices you should make for your overall database environment.
**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 Information/Management, 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. However, you may want to use multiple SDDS clusters for performance reasons.

Using a multiple-cluster SDDS does not change the way you use Information/Management. 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 utility programs such as BLGUT1 and BLGUT7. For information on the utility programs, refer to the *Operation and Maintenance Reference*.

For existing Information/Management databases, 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 *Operation and Maintenance Reference*.

For details on setting up a multiple-cluster SDDS, refer to the *Planning and Installation Guide and Reference*.

**SDDS Key Length**

The SDDS consists of one or more key-sequenced VSAM data sets. When defining the SDDS, your system administrator specifies the length of the key field in the SDDS VSAM record as 7 or 8 bytes for an Information/Management format database. Key 7 format generally provides better performance, and is the recommended format. Key 7 format is also required to perform some functions of Information/Management, such as backing up your database using the utility BLGUT23 or converting external date formats using the utility BLGUT17.

The SDDSSs for databases 0, 1, 2, and 3 must have a key length of 8.

Details on how the key is specified are provided in the *Planning and Installation Guide and Reference*.

**Multiple-Cluster SDIDS**

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 amount of data you can store in the SDIDS is 400GB.

Each SDIDS cluster contains only a subset of the entire data set. A multiple-cluster SDIDS can be useful to help improve your overall database performance because 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.

When evaluating whether or not you should set up a multiple-cluster SDIDS, you
should consider how records are updated in your organization (concurrently or
not), what kind of search contention will exist for those records in the same
cluster, the number of records that will exist per cluster, and how much read/write
activity is anticipated for the records.

For details on setting up a multiple-cluster SDIDS, see the Planning and Installation
Guide and Reference.

SDIDS Key Length

The SDIDS consists of one or more key-sequenced VSAM data sets. Your system
administrator can specify, when defining the SDIDS, that the key length be 18 or
34 bytes long.

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 DBCS characters on a search.

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

For more information on how the key length setting affects searching, refer to the
User’s Guide. Details on how the key is specified when the SDIDS is defined are
provided in the Planning and Installation Guide and Reference.

Determining Your Backup/Recovery Approach

Information/Management provides various ways for you to backup your data and
recover it if necessary. Depending on your specific needs, the size of your data
base, your desired frequency for backups, and your data base availability needs,
you may decide that one approach is better than another. The methods available to
you are:

- Use the BLGUT3 utility to restore the data base from the log data set created
  using BLGUT4 and pruned using BLGUT4LP. This method involves using the
  AMS REPRO command to create a backup of the SDDS and SDIDS. This
  method is commonly used and has been available for a number of years. The
  advantage of this method is that it is simple. The disadvantage is that the
  SDDS and SDIDS are unavailable while AMS REPRO copy is being created.

- Use the BLGUT23B utility to create a new master data set each time a backup
  is desired. The new master combined with a SDLDS large enough to hold all
  changes between runs of BLGUT23B form a complete backup copy of your
  data base. If it is necessary to restore the data base, BLGUT23R would be used
  after the SDLDS is offloaded with BLGUT4 and the one log processed using
  BLGUT23P.
The advantage of this is that the data base is always available and the processing steps are simple. This approach may be ideal for small organizations that use a single cluster SDDS that in the past has been backed up using IDCAMS repro.

The disadvantage of this is that the SDLDS may have to be very large to hold all the changes that occur to the data base between runs of BLGUT23B. Note, however, that a large SDLDS does not impact performance of Information/Management. Multiple–cluster SDDS users would have to use a sort merge process to merge masters created by BLGUT23B to create a complete master.

- Use the complete series of BLGUT23 utilities to create and merge data into a master backup data set. The advantage here is that your data base is available 24 hours a day, 7 days a week. Since BLGUT23B can take several hours to run each time, this method would avoid having to rerun BLGUT23B to create a new master. BLGUT23B would only be run once when the initial master is created. Then the master would be kept current using BLGUT23P and BLGUT23U after the SDLDS is offloaded periodically using BLGUT4. This backup method may be ideal for large enterprises with multiple SDDS clusters or high update activity which prevents the SDLDS from containing all the changes between backups. The disadvantage is the need to frequently merge the offloaded logs into the master using BLGUT23P and BLGUT23U.

- The Automatic Log Save facility periodically offloads the SDLDS and the offloaded log is then used to update another Information/Management data base. The Automatic Log Save facility enables organizations to maintain a complete shadow copy of their Information/Management data base. The advantages are:
  - The production data base is always available (24 by 7).
  - A production data base can be recovered very quickly by simply altering the SDDS and SDIDS cluster names of the backup data base to match the production data base. Or, after deleting the corrupted SDDS and SDIDS, you can use repro to replace the production data base with the backup data base. After altering the cluster names or completing the repro and processing any remaining offloaded logs, the data base could be placed back into production.
  - If the Automatic Log Save facility data base is defined in the session–parameters member as read–only, it can be used as reporting data base. This reduces the number of users on the production data base and should improve performance of the production data base.

The disadvantage of the Automatic Log Save facility is that it requires a duplicate SDDS and SDIDS.
Reuse of SDDS Position Numbers (Root VSAM Keys)

Each logical record (RNID) on a data base is assigned an SDDS position number (sometimes called a root VSAM key or a VSAM sequence number) to uniquely identify that record on the data base. When a new record is created, it is assigned a position number and stored in that location on the database. When the record is deleted, its SDDS position number becomes unassigned and its slot can be reused.

In earlier releases of Information/Management, SDDS position numbers were always reused. In Release 1.1, however, the default is to NOT reuse the numbers. When SDDS position numbers are reused, records are kept close together on the database, although not in any particular order. By not reusing the SDDS positions, new records are always added to the end of the database. Thus, whether you have a new database or an existing database, the SDDS position numbers are not reused unless you specifically request this option.

An advantage of not reusing the SDDS position number is that a search results list does not need to be sorted in order to display records in system-assigned RNID order. This should improve search performance by reducing contention for the SDDS because the search results list is already in system-assigned RNID order. On the other hand, if SDDS position numbers are not reused, any new records are always added to the end of the database, even though space may be available elsewhere where other records have been deleted. The unused space can be reclaimed by using IDCAMS to reorganize the SDDS whenever a significant number of records has been deleted.

The Set Database Options utility, BLGUT9, introduced in Release 1.1, allows you to select the option of reusing SDDS position numbers. See the discussion of BLGUT9 in the Operation and Maintenance Reference for additional information on this utility.

Creating Database Sorted by Record Number

You can use the BLGUT23 series of utilities to create a copy of an existing database. Then sort the copy of the database using RNID as the sort key. This sorted copy of the database can be used to create a database with the records in RNID order. If VSAM position numbers are not reused, then the database will stay in system-assigned RNID order. However, the order of user-assigned records will not be maintained.

Note: After performing this process you should not run BLGUT9.

Planning

- Your SDDS must use the key 7 format. If you are using the key 8 format, you must convert to the key 7 format using BLGUT7 before using this process. Your database will be unavailable while BLGUT7 runs.
- This procedure is not necessary for new (empty) databases, because they will start in system-assigned RNID order unless BLGUT9 is used to set the database option to reuse VSAM position numbers.
Your database can be used up to the time that you free the SDDS(s) and SDIDS(s). The SDLDS will contain the updates that occur from the time BLGUT23B starts until you actually begin freeing the SDDS and SDIDS.

You must run BLGUT1 as part of this process.

Automatic Log Save users must re-synchronize the Send and Receive databases after performing this procedure.

You may want to perform the following process on a small test database so that you can become familiar with the BLGUT23 series of utilities and perfect your JCL for merging (if you are using multiple SDDS clusters) and sorting of the master file that will be used as input to BLGUT23R.

Because your database will be kept in system-assigned RNID order, you should review any table panels used to display a search results list. If those table panels used the sort option of PREFIX and the prefix was RNID you may want to change them to use a sort option of DATABASE instead. See the section on Sorting the Search Results List in Panel Modification Facility Guide.

Procedure

To sort your existing database do the following:

1. Use BLGUT4 to offload the SDLDS. This offload load will not be used as part of creating the sorted database. However, it should be kept until BLGUT23B completes successfully.
2. Utilize BLGUT23B for each SDDS cluster. Users can continue to use the database while BLGUT23B is running.
3. If you have multiple SDDS clusters, merge the individual BLGUT23B backups into one master backup. See BLGUT23B in the Operation and Maintenance Reference for more information.
4. After BLGUT23B completes and you have merged the SDDS clusters (if you are using a multiple cluster SDDS), have your users stop using the database.
5. Free the SDDS(s) and SDIDS. Your users cannot access the database until the remainder of these steps are complete.
6. Run BLGUT4 to offload the SDLDS.
7. Run BLGUT23P to prune the offload log.
8. Run BLGUT23U to merge the pruned log with the master backup.
9. Sort the new updated master using the following sort control card

   \text{SORT FIELDS=(9,8,CH,A),EQUALS,SIZE=Ennnnnn}

   where \text{Ennnnn} is the total number of physical records in the master backup data set. Refer to the BLGUT23U SYSPRINT data set for the BLG21331I message. Typically, there are three physical records for each logical record. BLGUT20 will give you the actual average number of physical records of your data.
Creating Database Sorted by Record Number

- The above sort control card is for DFSORT. You may have to modify it if you use a different sort program.

10. After the sort successfully completes, use IDCAMS to delete your SDDS(s) and SDIDS(s). Then define new (empty) clusters using the same attributes as your existing clusters.

11. Reallocate the SDDS(s) and SDIDS(s) using the UTIL option on the BLX-SP REALLOC command.

12. Run BLGUT23R using the RESEQ parameter.

   **Note:** Do not use a BLGLOGUP input data set.

13. After BLGUT23R completes successfully, run BLGUT1 to rebuild the SDIDS.

14. After BLGUT1 completes successfully, reallocate the SDDS(s) and SDIDS(s).

15. Access the database using Information/Management; do a search and display the results on a table panel that has the sort option set to DATABASE. The records should appear in ascending or descending order depending on the sort order used on the table panel. (Ascending is the default.)

16. If the database appears the way you expect, you can allow your users to access the database. You might want to make a backup copy at this time using your preferred backup utility or BLGUT23B. If you use BLGUT23B, you can allow your users to access the database while it runs.

   **Note:** Automatic Log Save users should recreate their receive database by using the “Restoring synchronization without using BLGUT1” procedure documented in the *Program Administration Guide and Reference*.

17. Once this BLGUT23B successfully completes for all of your SDDS clusters (and the merge if necessary), discard the sorted master from which you created the sorted database and also discard any offloaded logs or older master backups.
Logical Database Partitioning

Logical Database Partitioning makes it possible for you to organize data in your Information/Management 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 Information/Management 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 records to logical partitions.

The database administrator controls which users have access to which partitions. This is done by specifying partition names in privilege class records.

Note: Before adding partition information to privilege classes, ensure that the BLGTSX DD is allocated and that the SBLMTSX data set is specified in the BLGTSX concatenation. Refer to the Planning and Installation Guide and Reference for more information on allocating BLGTSX.

Primary Partition Name

A Primary Partition Name can be added to a privilege class record. This Primary Partition Name identifies the logical partition for users in that privilege class, and thus the records, to which these users have access. The Primary Partition Name is assigned to every record that is created. Searches and other record access functions automatically access only the records in the primary partition.

Owning Partition Name

When a user enters a record into the Information/Management database, the Primary Partition Name from the current privilege class is collected as the Owning Partition Name in the record. Only users with access to that partition can access that record.

Note: If a privilege class record does not contain a Primary Partition Name, then database partitioning is not used and a user of this privilege class has access to every record in the database. Records are not assigned an Owning Partition if they are entered by a user using a privilege class that does not have a Primary Partition Name. This means that all such records in the database will be accessible to any users having a privilege class that does not contain a Primary Partition Name.
Secondary Partition List

If you are using Logical Database Partitioning and you want users in one partition to be able to access records in another partition, you can set up a Secondary Partition List in the users' privilege class record. The Secondary Partition List defines additional partitions that the privilege class can access. Users can access records in any of their secondary partitions; however, they cannot create any records in these partitions. A user cannot use record commands, such as UPDATE or DISPLAY to access a record in a secondary partition. Records in secondary partitions are only accessible through a search results list. See “Record Searches” on page 4-17 for information on how to access records in secondary partitions.

Authorizing a Privilege Class for Logical Database Partitions

This is a summary of how to set up your privilege class records for logical database partitioning:

- Create a privilege class record (or modify an existing privilege class record) and add a Primary Partition Name to the privilege class record. Any records entered into the database by users in that privilege class are assigned an Owning Partition Name corresponding to the Primary Partition Name from the privilege class.

- If the privilege class needs access to more than one partition, you can modify the privilege class record by adding a Secondary Partition List. This Secondary Partition List identifies partitions (in addition to the Primary Partition) that the privilege class can access. Records in secondary partitions are only accessible from a search results list.

The following panels illustrate how to create a privilege class containing a Primary Partition name and add Secondary Partitions to that privilege class.

Note: If a privilege class does not contain a primary partition name, database partitioning is not used. Users in that class have access to all records in the database.

Note: If the application specified in the upper-right corner of the panel is Management, change the setting to System.
From the Primary Options Menu, BLG0EN10, select ENTRY. Type 5 and press Enter.

To create a privilege class record, type 1 and press Enter.
On BLG0J100, the Class Description Entry panel, complete the required fields and any others you choose. In the **Primary partition id** field, type the Primary Partition Name. In this example, a Primary Partition Name of PARTIT01 is assigned to privilege class LEVEL01.

When you have entered the information, type `end` and press Enter.

```plaintext
===> end
```
On BLG0JU00, the Class Summary panel, select 4 to specify a list of secondary partitions accessible to a user in this privilege class. Type 4 and press Enter.

On BLGLJSPL, list other partitions which a user in this privilege class can access. In this example, users in privilege class LEVEL01 are given access to secondary partitions PARTIT02, PARTIT03, PARTIT04, and PARTIT05.

When you have completed the list, type end and press Enter.
You return to the Class Summary panel. To save the privilege class record, select **File record**. Type 9 and press Enter.

```
| User ID 1............ | TEMP99 | Owning priv. class....... |
| Transfer-to class..... | _______ | Entry priv. class....... |
| Contact name.......... | _______ | Date entered............. |
| Contact phone.......... | _______ | Time entered............. |
| Contact dept........... | _______ | Date last altered........ |
| Location code.......... | _______ | Primary partition....... |

Description............ BASIC PRIVILEGE CLASS
```

Select one of the following, type END to save your changes, or type CANCEL to discard your changes.

1. Class description.
2. Class authority.
3. Eligible users.
4. Secondary partition list.
5. Sharing partition list.

***=> 9***
Note that a message confirms that record LEVEL01, the privilege class record that you have created, was stored successfully.

```
BLGOEN10   --- PRIMARY OPTIONS MENU ---   APPLICATION: SYSTEM
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. DBADMIN........Perform database administration.

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

Privilege Class Selection

When a user selects a privilege class, the user is assigned the Primary Partition Name and any Secondary Partitions associated with that privilege class. If a user selects a different privilege class, the Primary Partition Name and Secondary Partitions associated with that privilege class are assigned.

A user can access the records in any partitions authorized by the privilege class. However, users can only create records in their primary partition; record commands and utility functions such as UPDATE or DISPLAY only access records in their primary partition. Records in secondary partitions are accessible from a search results list. By default, a search only searches records in a user's primary partition, but secondary partitions can be added to the search as described in “Record Searches” on page 4-17.

The Invocation Class in a user's profile designates a privilege class that the user automatically runs in when starting Information/Management. If the user wishes to change the Invocation Class in their profile, the chosen privilege class must contain the user's current Primary Partition as either its Owning Partition Name or as an entry in the Sharing Partition List (described in “Sharing Partition List” on page 4-13).

Although you cannot have duplicate RNIDs within a partition, you may have duplicate RNIDs across partitions. For that reason, care must be taken when specifying either the Invocation Class in a user's profile or the CLASS(NAME) on the ISPSTART command because only unique privilege class RNIDs are supported.
via these mechanisms. Duplicate privilege class RNIDs are supported via the Privilege Class selection list (Option 4 on the Primary Options menu).

**Universal Partition Access Authority**

A user can have access to all partitions, and therefore all records in the Information/Management database, when using a privilege class that has **Universal Partition Access** authority. A record entered by a user using a privilege class that has this authority is assigned the Primary Partition Name from the privilege class as its Owning Partition Name. However, record commands and utility functions such as UPDATE or DISPLAY access all records in the database. Because it is possible to have duplicate RNIDs across partitions, a user with Universal Partition Access authority can only access a record with a duplicate RNID by building a search results list that lists the record and working with the record from that list.
The following panels illustrate how to modify a privilege class record to assign it Universal Partition Access authority.

Update the Privilege Class record which you just created. Type `upd r level01` and press Enter.

From the Class Summary panel, type 2 and press Enter.
Because this is a System Authority, type 1 and press Enter.

On the Authority Entry panel, select **UNIVERSAL**. Type 4 and press Enter.
Now you are ready to give **Universal Partition Access** authority. Type `yes` and press Enter.

```
+ BLG6AUUA -- UNIVERSAL PARTITION ACCESS AUTHORITY LEVEL --- AUTH/+-
USE....Reply YES to indicate authority for this class.
NOTE...Reply ~ to remove universal access authority for this class. Reply ;BACK P to return to the options panel without modifying the authority.

REPLY: YES - The privilege class is allowed to access records in all partitions.
~ - Remove universal access authority from this class.
```

```===> yes
```

You return to the Authority Entry panel. Type `end,end` and press Enter.

```
+ BLG0J212 -------------- AUTHORITY ENTRY ----------------- 1 OF 1-+
SELECT THE TYPE OF AUTHORITY TO BE ENTERED
OPTIONS:
1. CLASS......Authority for privilege class records.
2. SRC........Authority for SRC (stored response chains).
3. DBADMIN....Authority for database administration.
4. UNIVERSAL..Authority to access all logical partitions.

+ Enter your choice, or type ;END or ;CANCEL to leave this panel. -+
```

```===> end,end
```
You return to the Class Summary panel. To save the privilege class record, select **File record**. Type 9 and press Enter.

A message confirms that record LEVEL01, the privilege class record that you have modified, was stored successfully.
Global Partition

If you have a set of records which you would like to be accessible to all users, put all such records into a “Global” Partition. Records in a Global Partition are accessible by all privilege classes.

To set up a global partition, you must define a Global Partition Name. It can be defined in either of two ways:

- Use program exit BLG01448 (described in Panel Modification Facility Guide) to define a Global Partition Name.
- Specify a value for the GBLPID keyword on the BLGPARMS macro in a session-parameters member. This keyword, which is optional, defines a Global Partition Name. A value of up to nine alphanumeric characters can be specified for the GBLPID keyword. The Planning and Installation Guide and Reference contains detailed information on the GBLPID keyword.

These are the rules for specifying a Global Partition Name:

- If a value is not provided for the GBLPID keyword in the session-parameters member and a Global Partition Name is not specified by program exit BLG01448, then no Global Partition exists.
- If a value is specified on the GBLPID keyword, that value is used instead of the value set by program exit BLG01448.
- If a value of GBLPID=' ' is specified, the Global Partition Name established by program exit BLG01448 is ignored.
- If the GBLPID keyword is omitted, the Global Partition Name set by program exit BLG01448 is used.

Once the Global Partition Name is defined, it can be assigned to a record as the Owning Partition in either of two ways:

- Add records to the database using a privilege class that has the Global Partition Name as its Primary Partition Name.
- Use program exit BLG01449 (described in the Panel Modification Facility Guide) to assign the Global Partition as the Owning Partition at the time the record is entered.

Sharing Partition List

The Owning Partition, once assigned to a record, must not be changed. Only if you have that record's Owning Partition Name as the Primary Partition Name in your privilege class can you use record access commands or utility functions to access the record. If you have the record's Owning Partition listed as a Secondary Partition in your privilege class, you can access the record via a search results list. If you also have the need to allow other partitions to have access to an individual record, you can set up a Sharing Partition List for that record. This Sharing Partition List permits an individual record to be shared by multiple partitions in addition to that identified by the record's Owning Partition Name. When the
Primary Partition Name in your privilege class is listed in the Sharing Partition List for an individual record that is owned by a different partition, you have access to that record as though it were in your primary partition.

Information/Management allows you to enter a list of Sharing Partitions for privilege class records. If you need to be able to enter Sharing Partitions for other types of records, you will need to use the Panel Modification Facility (PMF) to customize your panels. “PMF Considerations” on page 4-20 provides some characteristics about Sharing Partitions that you will need to know before you customize your system.

The following example shows you how to add Sharing Partitions to a privilege class record.

Update the privilege class record with which we have been working by typing `upd r level01` and press Enter.
To create a Sharing Partition List which identifies other partitions that share this record, type 5 and press Enter.

For privilege class records, the Primary Partition Name from the privilege class being updated or created is automatically added to the Sharing Partition List. You can enter the names of additional partitions that share this record. In this example, PARTA, PARTB, and PARTC are added to the Sharing Partition list. When you have completed the list, type end and press Enter.
The Class Summary panel summarizes the information about this privilege class record. To save the privilege class record, select **File record**. Type 9 and press Enter.

```
<table>
<thead>
<tr>
<th>BLGOJU00</th>
<th>CLASS SUMMARY</th>
<th>CLASS: LEVEL01</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID 1.............. TEMP99</td>
<td>Owning priv. class....</td>
<td></td>
</tr>
<tr>
<td>Transfer-to class.......</td>
<td>Entry priv. class.....</td>
<td>MASTER</td>
</tr>
<tr>
<td>Contact name...............</td>
<td>Date entered........</td>
<td>07/16/96</td>
</tr>
<tr>
<td>Contact phone...............</td>
<td>Time entered.........</td>
<td>12:16</td>
</tr>
<tr>
<td>Contact dept...............</td>
<td>Date last altered....</td>
<td>07/16/96</td>
</tr>
<tr>
<td>Location code.............</td>
<td>Primary partition.....</td>
<td>PARTIT01</td>
</tr>
<tr>
<td>Description...............</td>
<td>BASIC PRIVILEGE CLASS</td>
<td></td>
</tr>
</tbody>
</table>
```

Select one of the following, type END to save your changes, or type CANCEL to discard your changes.

3. Eligible users.  
4. Secondary partition list.
5. Sharing partition list.

```text
>>> 9
```

A message confirms that record LEVEL01, the record for which you have added Sharing Partition information, was stored successfully.
Logical Database Partitioning Considerations

Unique RNIDs by Partition

There is a record in each partition which contains the last system-assigned record number (also referred to as the last entry number) for that partition. This allows a unique value to be specified for the last entry number for each partition. There is also a record which contains the last entry number for non-partitioned records. System-assigned RNIDs are assigned based on the appropriate last entry number record from the SDIDS. If the user's privilege class has a Primary Partition Name, the partition's last entry number record is used when assigning a system-assigned RNID. Otherwise, the database last entry number record is used.

For user-assigned RNIDs, Information/Management checks only the Primary and Global Partitions for duplicate RNIDs.

Record Searches

In general, when a user issues the SEARCH command, only records in the user's primary partition are returned. If a Global Partition is defined, then the records owned by the Global Partition are also returned. In addition, if you have customized Information/Management to collect Sharing Partitions for individual records, those record are returned if you are authorized.

A user's privilege class also identifies secondary partitions that the user has authority to access. Records in secondary partitions are only accessible from a search results list. Databases 5, 7, 8, and 9 support secondary partitions. To add secondary partitions to a search, you must use the SEARCH command and specify the \( n^* \) operand (where \( n \) is the database identifier; for example, \( 5^* \)). The \( n^* \) operand on the SEARCH command causes all of the secondary partitions to which you have access to be searched.

Program exit BLG01450 (described in the Panel Modification Facility Guide) can be used to force the next search operation to be a multipartition search.

Note: API applications cannot perform multipartition searches because an API is unable to retrieve a record by its VSAM root key.

Because a search can access multiple partitions, and because it is possible to have records in your search results list with duplicate RNIDs, you may want to know the partition to which a particular record belongs before you access that record from a search results list. Your Search Results List panels can be customized to show the Owning Partition Name for each record in the list. A Search Results List panel BLGISRP that displays the Owning Partitions Names is shipped by Information/Management as an example.
Logical Database Partitioning Considerations

Copying Records between Partitions
When a record is copied from one partition to another, the Owning Partition Name in the copied record is set to the Primary Partition Name from the current privilege class. You can use program exit BLG01151 to delete the Sharing Partition List from a record when it is copied by adding the list processor root s-word associated with the Sharing Partition List to BLG1ACOP. “PMF Considerations” on page 4-20 contains the Sharing Partition List root s-word.

Record Ownership
Each record in the database contains three privilege class names: the Entry Class, the Owning Class, and the Transfer-to Class.

The Owning Class field in a record describes its privilege class ownership. Record ownership can be transferred across partitions by changing the Transfer-to Class field in the record. In order for the Transfer-to Class receiver to be able to access the record, the record whose ownership is being changed must contain the Primary Partition Name of the Transfer-to Class receiver either as its Owning Partition Name or as a member of its Sharing Partition List. In addition, the privilege class of the Transfer-to Class receiver must contain the Owning Partition Name of the record either as its Primary Partition or as a member of its Secondary Partition List.

It is not possible to differentiate between like-named privilege classes in different partitions.

Commands in a Logical Partition Environment
Although you cannot have records with duplicate RNIDs within a partition, records in different logical partitions may have duplicate RNIDs. If you are authorized by your privilege class (by Universal Partition Access or by Sharing Partitions) to records in more than one partition that have duplicate RNIDs, you cannot use Information/Management record commands or utility functions (such as UPDATE or DISPLAY) to access these records. However, those records are accessible from a search results list. A search that includes secondary partitions may also result in a search results list that has records with duplicate RNIDs. Line commands can be used to access any record in the search results list.

The COPY, DISPLAY, PRINT, DELETE, and UPDATE Commands
The COPY, DISPLAY, PRINT, DELETE, and UPDATE commands retrieve the target record by issuing a search for the specified RNID. Only records with unique RNIDs can be processed by these commands and their corresponding utility functions.

In general, these commands access records in the user's primary partition (and Global Partition), in which case your records will have unique RNIDs. If your privilege class has Universal Partition Access authority or you use Sharing Partitions, it may be possible for you to use these commands or utility functions to access records with unique RNIDs in other partitions. If you have access to
records with duplicate RNIDs, you must use a search results list in order to access these records.

The GLOSSARY Command
The GLOSSARY command shows only information relevant to the Primary, Secondary, and Global Partitions, unless the current privilege class has Universal Partition Access or MASTER authority. MASTER authority is required to display the last entry number values, including the value for LASTENTRYNUMBER.

The ORDER Command
Because Information/Management does not apply database partitioning to the ORDER command, you should use the command alias and authorization function (described in “Defining Command Aliases and Authorization” on page 9-1) to restrict its usage.

The STATISTICS Command
Because Information/Management does not apply database partitioning to the STATISTICS command, you should use the command alias and authorization function (described in “Defining Command Aliases and Authorization” on page 9-1) to restrict its usage.

Migrating Existing Records to a Logical Partition
Existing records can be logically partitioned via a user-written TSP.

The TSP must perform the following processing:

- Execute a search to create a Search Results List (SRL) containing records that are to be grouped into a logical partition.
- Block update the records in the SRL.
- Use the TSP WORDFIX control line to add the Owning Partition s-word and p-word, including the partition name, to each record in the list.

Note: WORDFIX is not supported in a TSX.

For additional information on how to add data to a record using the TSP WORDFIX control line, refer to the Terminal Simulator Guide and Reference. An example of how to perform a search and block update of a group of records can also be found in that document.

Note: You can damage your existing database if you do not use the WORDFIX control line correctly. For information on the security measures you can use to protect against its misuse, refer to the discussion of data integrity and security using TSPs in the Planning and Installation Guide and Reference.

Each partition will have a record in the SDIDS, similar to the database LASTENTRYNUMBER record, to keep track of the last system-assigned RNID in that partition. The key of the last entry number for a partition has the following format:
PTID/ptidname

When the first record with a system-assigned RNID is filed in a partition and the last entry number record for the partition does not exist in the SDIDS, the following occurs:

- A last entry number record is created for the partition.
- The initial RNID for the partition last entry number record will be 00000001 if there is no record in the SDIDS with that PTID/ptidname or there is no database LASTENTRYNUMBER record. Otherwise it will be the value of LASTENTRYNUMBER. A user with MASTER authority can use the GLOSSARY command to inspect the last entry number records for the database and its partitions if they exist.

**PMF Considerations**

If you are using logical database partitioning, you should know that the p-word PTID/ is used by Information/Management to designate partition identifiers. Your program administrator should check your dictionary to see if your installation is using PTID/. If so, your administrator should use program exit BLG01448 to change the default p-word for the partition.

The following table contains a list of the s-words and p-words (and their major characteristics) associated with Information/Management logical database partitioning.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>s-word</th>
<th>s-word index</th>
<th>p-word</th>
<th>p-word index</th>
<th>prefix validation pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owning Partition</td>
<td>XIM0I0CPO</td>
<td>X'0174'</td>
<td>PTID/</td>
<td>X'03C9'</td>
<td></td>
</tr>
<tr>
<td>Sharing Partition List</td>
<td>XIM0I0CH</td>
<td>X'011D'</td>
<td>PTID/</td>
<td>X'011D'</td>
<td>SSV8</td>
</tr>
<tr>
<td>Primary Partition</td>
<td>XIM0I0CPP</td>
<td>X'011B'</td>
<td>X'03C7'</td>
<td>SSV8</td>
<td></td>
</tr>
<tr>
<td>Secondary Partition List</td>
<td>XIM0I0CS</td>
<td>X'0118'</td>
<td>X'03C7'</td>
<td>SSV8</td>
<td></td>
</tr>
</tbody>
</table>

**Predefined Validation Patterns**

Equal sign response validation patterns that are available are described in the *Panel Modification Facility Guide*. 
Archiver Considerations

If you are using the archiver function (described in “The Information/Management Archiver” on page 15-1), you must be aware of the restrictions described in “Logical Database Partition Considerations” on page 15-35.
Customizing the Enhanced Panel Style

This information is provided for the program administrator who is responsible for customizing the enhanced panel style for Information/Management. To do the tasks explained in this chapter, you must be familiar with modifying ISPF panels and tables. For more information, refer to the following publications for the prerequisite version of ISPF:

- ISPF User’s Guide
- ISPF Dialog Developer’s Guide and Reference
- ISPF Edit and Edit Macros
- ISPF Dialog Tag Language and Reference
- ISPF Planning and Customizing

Refer to the ISPF Dialog Developer’s Guide and Reference for information about enabling applications for ISPF graphical user interface (GUI) mode.

Information/Management Panel Styles

Information/Management provides the following panel styles:

- The standard panel style, which provides a command line. See “Standard Panel Style” on page 5-2 for an illustration.
- The enhanced panel style, which provides an action bar, pull-down menus, and can display context-specific function keys, accelerator keys, and separator bars. A set of enhanced panel style windows is provided in the sample data set. See “Enhanced Panel Style” on page 5-3 for illustrations. Refer to the User’s Guide for information on using the enhanced panel style.

Information/Management provides the WINDOW command to enable you to change the panel style you are using. If you use the enhanced panel style, you can choose between automatic window selection and manual window selection. Refer to the User’s Guide for command syntax and details.
Standard Panel Style

This is the Primary Options Menu displayed using the standard panel style window, BLGISPFD.

BLGOEN20   --- 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 Information/Management panels.

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

The following enhanced panel style windows are provided:

This is the Primary Options Menu displayed using the enhanced panel style window BLGISPFE, the primary window.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Dialog</th>
<th>Record</th>
<th>Window</th>
<th>Options</th>
<th>Help</th>
</tr>
</thead>
</table>
--------------------------------------------------------------------------------------------------------------------------
BLGOEN20 --- 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 Information/Management panels.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997
This is a search results list displayed using the enhanced panel style window BLGISPFI, the inquiry window.

<table>
<thead>
<tr>
<th>DATABASE: 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECORD ID</td>
</tr>
<tr>
<td>1. FIRSTONE</td>
</tr>
<tr>
<td>2. MASTER</td>
</tr>
<tr>
<td>3. TESTER</td>
</tr>
<tr>
<td>4. OTHERS</td>
</tr>
<tr>
<td>5. 00000596</td>
</tr>
<tr>
<td>6. 00000597</td>
</tr>
<tr>
<td>7. 00000598</td>
</tr>
<tr>
<td>8. 00000599</td>
</tr>
<tr>
<td>9. 00000600</td>
</tr>
<tr>
<td>10. 00000601</td>
</tr>
<tr>
<td>11. 00000602</td>
</tr>
<tr>
<td>12. 00000603</td>
</tr>
<tr>
<td>13. 00000604</td>
</tr>
<tr>
<td>14. 00000605</td>
</tr>
</tbody>
</table>

Line Cmds: C=Copy D=Delete P=Print S=Select U=Update
Type DOWN or UP to scroll the panel, or type END to exit the panel.

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
Enhanced Panel Style

Information/Management Panel Styles

This is an example of panel externals displayed when updating a panel in PMF. This display is using the enhanced panel style window BLGISPFA, the administration window.

As a program administrator, you may want to modify one or more of the enhanced panel style windows to meet the requirements of your users. For example, you may want to:

- Enable the windows in the enhanced panel style set for ISPF GUI mode
- Change one of the windows in the enhanced panel style set
- Create additional windows
- Change the function key definitions

If your organization has installed the enhanced panel style and some of your users want to use the standard panel style, you can accommodate that group of users. See “Changing the Default ISPF Panel” on page 5-8 for details.

Understanding the ISPF Panels

From a program administrator’s point of view, Information/Management windows are actually ISPF panels. The ISPF panels that you use must reside in a data set that is in your ISPPLIB concatenation. Refer to the Planning and Installation Guide and Reference for installation instructions. The following ISPF panels are shipped in the SBLMSAMP data set:
Understanding the ISPF Panels

**BLGISPFD** Contains the standard panel style
**BLGISPFE** Contains the primary window for the enhanced panel style
**BLGISPF1** Contains the inquiry window for the enhanced panel style
**BLGISPFA** Contains the administration window for the enhanced panel style

You must also have ISPF panel BLGISPFM in a data set that is in your ISPPLIB concatenation. This panel contains the default panel style for your installation. This style was chosen at the time Information/Management was installed. It is a copy of either BLGISPFD or BLGISPFE.

**ISPF Tables**

Because Information/Management commands and ISPF commands can have the same name, Information/Management provides the following ISPF tables:

**BLG0CMDS** Contains a mapping of alias names for Information/Management commands to the proper Information/Management syntax.
**BLG0KEYS** Contains the keylists that map the alias names to function keys.

If you use the enhanced panel style, the data sets containing these tables must be in your ISPTLIB concatenation. Refer to the *Planning and Installation Guide and Reference* for installation instructions.

**Variables for Automating Window Selection**

The ISPF panels that display the enhanced panel style windows use the following variables to determine which of the enhanced windows to display:

**BLG#PTYP** Set to the type of the Information/Management panel. (Refer to the *Panel Modification Facility Guide* for descriptions and illustrations of Information/Management panel types.) The following values are used:

- **pmf** Specifies a PMF panel type
- **select** Specifies a selection panel type
- **assisted** Specifies an assisted-entry panel type
- **data** Specifies a data-entry panel type
- **table** Specifies a table panel type
- **other** Specifies that the panel type does not match any of the other values. The panel could be a control panel or a terminal simulator panel.

If the value of **BLG#PTYP** is **pmf**, the administration window (BLGISPFA) is displayed. If the value is **table**, the values of the next two variables determine whether to display the inquiry window or the primary window.

**BLG#PNAM** Set to the panel name. This variable is used to determine whether to display the primary window or the inquiry window when the value of **BLG#PTYP** is **table**. The inquiry window is displayed if the value of **BLG#PNAM** is one of the following:

- The same as the value of **BLG#TNAM**
- **BLG1TSRL**
- **BLG1TMSQ**
Enhanced Panel Style

Understanding the ISPF Panels

- **BLG1TSQ**

  If the value is anything else, the primary window is displayed.

- **BLG#TNAM**

  Set to the name of the Information/Management panel that is used to display a search results list. The following sources are used, in the order listed, to set this variable:

  1. The last TABLE command issued during this session
  2. The profile setting for the **Default panel** field in the Search options part of the Session Control panel

  The WINDOW command can be used to override automatic window selection. Refer to the *User’s Guide* for details.

Enabling the Enhanced Panel Style for ISPF GUI

To use the graphical user interface (GUI), you must have the correct version of ISPF installed and running on your system.

- To enable GUI mode for OS/2 or Microsoft Windows, you must use ISPF Version 4.1 or later.
- To enable GUI mode for AIX or HP, you must use ISPF Version 4.2 or later.
- To enable GUI mode for SUN SOLARIS, WINDOWS NT, or WINDOWS 95, you must use ISPF Version 4.2.1 or later.

Push buttons, mnemonic choice selections, and unavailable pulldown choices are supported automatically. The ISPF panels shipped with Information/Management that provide the enhanced panel style contain special keywords that exploit additional ISPF Version 4.2 graphical user interface enhancements. However, these keywords are disabled by comment lines because ISPF Version 4.1, which is the prerequisite level for Information/Management Version 1.1, does not support them.

If you are using ISPF Version 4.2 or later, you can remove the comment delimiters by following the instructions found within the enhanced panel style ISPF panels.

Removing the V42 comment delimiters enables these user interface controls when running Information/Management in GUI mode of ISPF Version 4.2 or later. Enabling these controls has no effect in 3270 mode.

**Accelerator keys**

This feature enables users to press a combination of keys to start a function from a pull-down menu. If an action item on a pull-down menu has an accelerator defined, you can start that action by using the accelerator—even if you have not switched to the action bar or displayed the pull-down menu. Refer to the *User’s Guide* for information on using accelerator keys.

**Separator bars**

This feature provides users with a visual distinction between adjacent areas within a pull-down menu.
The specific ISPF panels that are shipped with Information/Management that you must change are:

- BLGISPFE
- BLGISPFI
- BLGISPFA
- BLGISPFM, if it was created by copying BLGISPFA, BLGISPFE, or BLGISPFI.

If you have created other ISPF panels to run with Information/Management that you want to enable for these enhancements, refer to the ISPF Dialog Developer’s Guide and Reference for more information.

### Changing the Default ISPF Panel

The default ISPF panel is the one that is used for the first panel after initializing Information/Management.

To change the default ISPF panel for a set of users, copy the desired ISPF panel into BLGISPFM in the first data set in the users’ ISPPLIB concatenation.

An individual user can use the WINDOW command to select a default ISPF panel. For example an Information/Management panel developer can change the default ISPF panel to administration with the following command:

```plaintext
WINDOW BLGISPFA,WINDOW SAVE
```

Refer to the User’s Guide for more information on using the WINDOW command.

### Changing the Code for an ISPF Panel

You can change the code in an ISPF panel that displays an enhanced panel style window to:

- Add new actions to the action bar and corresponding pull-down choices
- Change the shipped action bar actions or corresponding pull-down choices
- Change the function keys.

See “Changing Keylists” on page 5-17 for information on changing the function keys.

To do the first two tasks, locate the ISPF panel you want to change. For this example, change BLGISPFE stored in INFO.SPECIAL.PANELS. Note that BLGISPFE is an ISPF panel, not an Information/Management panel. Therefore, you cannot use PMF to make these changes. You must update this panel in the way you usually update data sets. For more information on updating ISPF panels, refer to the ISPF Dialog Developer’s Guide and Reference.

**Note:** If you are using ISPF Version 4.2 or later, you can change the panels using ISPF Dialog Tag Language. See “Using DTL to Change an Action Bar” on page 5-16 for more information.
Adding an Action to the Action Bar

In this scenario, you want to add an action called TSO to the action bar. The pull-down menu for TSO has three choices: Time, Mail, and Profile. Time is the default choice.

This is how ISPF panel BLGISPFE appears when you select it as your panel style. This is the way the panel is shipped with Information/Management.

Refer to the User's Guide for instructions on using the enhanced panel style.
This is how the code for panel BLGISPFE appears on your screen when you are in PDF edit mode. This is the way the panel is shipped with Information/Management; it has not been completely enabled for GUI mode.

Locate the line that begins with )BODY.

```
EDIT ---- INFO.SPECIAL.PANELS(BLGISPFE)---- 01.00  ---------  COLUMNS 001 072
COMMAND ===> SCROLL ===> PAGE
****** ********** TOP OF DATA **********
000001 )PANEL KEYLIST(BLGKEYLE,BLG0)
000002 )PANEL KEYLIST(BLGKEYLE,BLG0)
000003/**************************************************************************/
000004/* The following global change can be made to this panel   *
000005/* to exploit ISPF Version 4.2 user interface enhancements. *
000006/* *
000007/* If you are running ISPF Version 4.2:  *
000008/* Remove all comment prefixes of the form /*V42 by using the PDF  *
000009/* editor command,  C /*V42 '' ALL  *
000010/* *
000011/* WARNING: ONCE THESE CHANGES HAVE BEEN MADE, THIS PANEL WILL NOT  *
000012/* BE USABLE WITH RELEASES OF ISPF PRIOR TO VERSION 4.2. You may  *
000013/* want to save a copy of this panel before making the changes.  *
000014/**************************************************************************/
000015)ATTR FORMAT(MIX)
000016   | AREA(DYNAMIC) EXTEND(ON) DATAMOD(20)
000017  01 TYPE(DATAOUT) INTENS(LOW)
000018  02 TYPE(DATAOUT) INTENS(HIGH)          PAS(ON)
000019  03 TYPE(DATAOUT) INTENS(HIGH)
000020  08 TYPE(DATAOUT) SKIP(ON)
000021  09 TYPE(DATAOUT) INTENS(LOW) SKIP(ON)
000022  08 TYPE(DATAOUT) INTENS(HIGH) SKIP(ON)
000023  11 TYPE(DATAIN) INTENS(LOW) CAPS(OFF)
000024  13 TYPE(DATAIN) INTENS(HIGH) CAPS(OFF)
000025  15 TYPE(DATAIN) INTENS(LOW) CAPS(_OFF)
```
On the line following the one that begins with )BODY, insert the text that you want
to add to the action bar. Be sure to include attribute characters that tell ISPF how
to display the text. For this example, you want **TSO** before **Help**, so insert this text
before `Help`:

```
˜ TSO
```

Now you must add )ABC, PDC, and ACTION statements to create the pull-down
menu. The )ABC statement must include exactly the same text that you just
inserted before **Help**, but without the attribute characters. The PDC statements
identify the choices on the pull-down menu. The ACTION statements say what to
do when the user selects a particular choice.

The value of the DESC parameter on each PDC statement must be the exact text
that you want to appear in the list on the pull-down menu. If the text contains
multiple words, enclose the text in quotation marks. If the action is to run an
Information/Management command, the RUN() value must be BLGxxxxx (maximum
of 8 characters) as defined in the BLG0CMDS table. If the action to be run requires
parameters, use the PARM operand to pass them. For example, the ACTION
statement for **Time** would be ACTION RUN(TSO) PARM(TIME). Use quotes when the
PARM operand contains multiple parameters that are separated by spaces, for
example ACTION RUN(TSO) PARM("LISTBC MAIL NONOTICES").
For this example, the \(\text{\textbackslash ABC DESC value is TSO. Each choice on the pull-down menu requires a PDC statement. And each PDC statement requires an ACTION statement to specify what to run when each choice is made.}\)

Add these statements before the corresponding \(\text{\textbackslash ABC section for Help. The statements are illustrated in the following panel. These statements were added by copying and modifying the corresponding \(\text{\textbackslash ABC section for Help. If you decide to enable the panel for GUI mode, verify that the values of the MNEM parameters on the PDC statements you add are appropriate. Refer to the ISPF Dialog Developer’s Guide and Reference for more information.}\)

```
EDIT ---- INFO.SPECIAL.PANELS(BLGISPFE)---- 01.00 ---------- COLUMNS 001 072
COMMAND ===> SCROLL ===> PAGE
000185)ABC DESC(TSO) MNEM(1)
000186 PDC DESC(Time) MNEM(1)
000187 ACTION RUN(TSO) PARM(TIME)
000188 PDC DESC(Mail) MNEM(1)
000189 ACTION RUN(TSO) PARM('LISTBC MAIL NONOTICES')
000190 PDC DESC(Profile) MNEM(1)
000191 ACTION RUN(TSO) PARM(PROFILE)
000192)ABC DESC(Help) MNEM(1)
000193 PDC DESC('General Help') MNEM(1)
000194 ACTION RUN(BLGHEL)
000195 PDC DESC('Help Status') MNEM(1)
000196 ACTION RUN(BLGHESL)
000197 PDC DESC('Database glossary') /*V42 PDSEP(ON)
000198 ACTION RUN(BLGGLO)
000199)ABCINIT
000200 .ZVARS = ZPDC
000201 &ZPDC = ' '
000202 )BODY EXPAND(//) WIDTH(&BLG#DWID);
000203 " Environment " Dialog " Record " Window " Options " TSO" Help +
000204 " "&MORE +
000205 ]BLG#ILSD / / |
000206 )INIT
000207 .CURSOR = BLG#ILSD
000208 .CSRPOS = &BLG#COFF;
000209 .ALARM = &BLG#ALRM;
```
Follow the )ABC section you just added with an )ABCINIT section that duplicates the ZVARS and ZPDC assignment statements that are included on the other actions. The value that you assign to the ZPDC variable is the default choice value. In this example, assign a value of 1 to ZPDC to make Time the default choice. Add these statements before the )ABC section for HELP.

```
EDIT ---- INFO.SPECIAL.PANELS(BLGISPFE)---- 01.00 -------------- COLUMNS 001 072
COMMAND ===> SCROLL ===> PAGE
000185 )ABC DESC(TSO) MNEM(1)
000186 PDC DESC(Time) MNEM(1)
000187 ACTION RUN(TSO) PARM(TIME)
000188 PDC DESC(Mail) MNEM(1)
000189 ACTION RUN(TSO) PARM('LISTBC MAIL NONOTICES')
000190 PDC DESC(Profile) MNEM(1)
000191 ACTION RUN(TSO) PARM(PROFILE)
000192 )ABCINIT
000193 .ZVARS = ZPDC
000194 &ZPDC = 1
000195 )ABC DESC(Help) MNEM(1)
000196 PDC DESC('General Help') MNEM(1)
000197 ACTION RUN(BLGHEL)
000198 PDC DESC('Help Status') MNEM(1)
000199 ACTION RUN(BLGHELST)
000200 PDC DESC('Database glossary') MNEM(1) /*V42 POSEP(ON)
000201 ACTION RUN(BLGGLO)
000202 )ABCINIT
000203 .ZVARS = ZPDC
000204 &ZPDC = ' '
000205 )BODY EXPAND(//) WIDTH(&BLG#DWID);
000206 ` Environment ` Dialog ` Record ` Window ` Options ` TSO ` Help +
000207 `/-&MORE +
000208 |BLG#ILSD / /|
000209 )INIT
```
You can save the ISPF panel as BLGISPFE or save it under a different name. If you plan to save it under a different name, you must also update the statements that enable automatic selection of this window. See “Variables for Automating Window Selection” on page 5-6 for more information about these variables. For this example, save the panel as BLGISPFT. Assume that some of your users plan to use this as their primary window, so change all occurrences of BLGISPFE to BLGISPFT in this ISPF panel and in the other ISPF panels that automatically flow to this one.

```
EDIT ---- INFO.SPECIAL.PANELS(BLGISPFT)---- 01.00 -------------- COLUMNS 001 072
 00020  IF (&BLG#WCTL; = 'AUTO')
 00021     IF (&BLG#PTYP; = 'PMF')
 00022     &WINDOW = 'BLGISPFA'
 00023 ELSE
 00024     IF (&BLG#PTYP; = 'TABLE')
 00025     IF (&BLG#PNAM; = &BLG&BLG#TNAM,'BLGISRSL', 'BLGISMSQ', 'BLGISSQL')
 00026     &WINDOW = 'BLGISPFI'
 00027 ELSE
 00028     &WINDOW = 'BLGISPFT'
 00029 ELSE
 00030     &WINDOW = 'BLGISPFT'
 00031     IF (&WINDOW = &BLG#PRIM);
 00032     &BLG#PRIM; = &WINDOW
 00033     VPUT (BLG#PRIM) SHARED
 00034 .RESP = ENTER
 00035 &TEMP = LVLINE(BLG#ILSD)
 00036 IF (&BLG#DDEP; = &TEMP);
 00037 ELSE
 00038 .RESP = ENTER
 00039 &BLG#DDEP; = &TEMP
 00040 )PROC
 00041 &BLG#CSRN; = .CURSOR
 00042 &BLG#COFF; = .CSRPOS
 00043 VPUT (BLG#CSRN BLG#COFF) SHARED
 00044 &V = TRUNC(&ZENVIR,7);
```
If you copy BLGISPFT into BLGISPFM, you can make it the default for a group of users. Or an individual user can select it with the WINDOW command. Regardless of how the panel is selected, the adjacent window appears.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Dialog</th>
<th>Record</th>
<th>Window</th>
<th>Options</th>
<th>TSO</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG/zerodotEN2/zerodot --- PRIMARY OPTIONS MENU --- APPLICATION: MANAGEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 Information/Management panels.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997

If you want the TSO action to appear on the other windows in the enhanced panel style set (BLGISPFA and BLGISPFI), change those panels the same way you just changed BLGISPFE.

**Changing the Action Bar Action or Pull-Down Choice**

If you want a change to apply to more than one window, you must change each ISPF panel that you want to reflect the change. For example, if you want a change to appear only in the primary window, change only ISPF panel BLGISPFE. If you want a change to appear in the primary window and the inquiry window, change both BLGISPFE and BLGISPF.

To change an existing action bar item, change the text on the line following the line that begins with )BODY and change the corresponding )ABC DESC statement. For example, to change the text of Environment to Profile, locate Environment in both places and replace it with Profile.

To change the text of a pull-down choice, change the text in the appropriate PDC DESC statement. For example, to change the text Quit to Bye, locate Quit and replace it with Bye.
To change the default choice for action, change the ZPDC assignment value within the appropriate )ABCINIT section. For example, to change the default Dialog choice from Recall to End, change the value of ZPDC in )ABC DESC(Dialog) section from 1 to 3.

To delete an action, delete all the lines beginning with the )ABC statement that contains the action you want to delete and delete the lines in the corresponding )ABCINIT section. You must also delete the action from the line following the line that begins with )BODY.

To delete an individual choice from a pull-down menu, delete the two lines that contain the PDC and the ACTION statements. Note that this changes the ordinal choice number of any choices that follow the deleted choice. Therefore, if the ZPDC default was set for a following choice, decrement the value assigned to ZPDC.

For more information on changing ISPF panels, refer to the following publications:

- *ISPF Dialog Management Guide and Reference*
- *ISPF/PDF Edit and Edit Macros*
- *ISPF/PDF Guide and Reference*

**Using DTL to Change an Action Bar**

If you are using ISPF Version 4.2 or later, you can change the DTL source files for the ISPF panels, action bars, command table, and keylists instead of actually changing each individual ISPF panel. The source files are included in the sample library (SBLMSAMP) as follows:

<table>
<thead>
<tr>
<th>GML Part</th>
<th>Source For:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGIGMLA</td>
<td>Panel BLGISPFA</td>
</tr>
<tr>
<td>BLGIGMLD</td>
<td>Panel BLGISPFD</td>
</tr>
<tr>
<td>BLGIGMLE</td>
<td>Panel BLGISPF</td>
</tr>
<tr>
<td>BLGIGMLI</td>
<td>Panel BLGISPI</td>
</tr>
<tr>
<td>BLGABDBG</td>
<td>Debug action bar</td>
</tr>
<tr>
<td>BLGABDIA</td>
<td>Dialog action bar</td>
</tr>
<tr>
<td>BLGABENV</td>
<td>Environment action bar</td>
</tr>
<tr>
<td>BLGABHLP</td>
<td>Help action bar</td>
</tr>
<tr>
<td>BLGABOPT</td>
<td>Options action bar</td>
</tr>
<tr>
<td>BLGABPMF</td>
<td>PMF action bar</td>
</tr>
<tr>
<td>BLGABREC</td>
<td>Record action bar</td>
</tr>
<tr>
<td>BLGABSCR</td>
<td>Scroll action bar</td>
</tr>
<tr>
<td>BLGABSER</td>
<td>Search action bar</td>
</tr>
<tr>
<td>BLGABSRC</td>
<td>SRC action bar</td>
</tr>
<tr>
<td>BLGABWIN</td>
<td>Window action bar</td>
</tr>
<tr>
<td>BLGIATTR</td>
<td>Common panel )ATTR section</td>
</tr>
<tr>
<td>BLGIINIT</td>
<td>Common panel )INIT section</td>
</tr>
<tr>
<td>BLGIPROC</td>
<td>Common panel )PROC section</td>
</tr>
<tr>
<td>BLG0EPSC</td>
<td>Information/Management command table</td>
</tr>
<tr>
<td>BLG0EPSK</td>
<td>Information/Management keylist table</td>
</tr>
</tbody>
</table>
Use the ISPF/PDF editor to modify these source files. Refer to the *ISPF Dialog Tag Language Guide and Reference* for complete information about changing the tags in these files.

### Adding a New Action Bar

To add a new action bar definition to the BLGISPFE panel using the GML source files, follow these steps:

1. Create a new GML source file for the new action bar definition and use these tags: `AB`, `ABC`, and `PDC`.
2. Update the BLGIGMILE source file (for BLGISPFE) by adding the include statement for the new action bar definition.
3. Use the DTL conversion utility (ISPDTLC) to convert the BLGIGMILE source file to an ISPF table.
4. Update the BLG0EPSC source file (for the Information/Management command table, BLG0CMDS) by adding any new commands.
5. Use ISPDTLC to convert the BLG0EPSC source file to the command table, BLG0CMDS.
6. Test your modifications.
7. Move the modified BLGISPFE panel to your production ISPPLIB data set.
8. Move the modified BLG0CMDS command table to your production ISPTLIB data set.

### Changing an Existing Action Bar

The steps required to change an action bar definition are similar to those for adding a new definition when you use the GML source files. For this example, you will see how to change the *Options* definition.

1. Update the BLGABOPT source file (for the *Options* definition).
2. Use ISPDTLC to convert the BLGIGMLA, BLGIGMLE, and BLGIGMLI source files.
3. Update the BLG0EPSC source file (for BLG0CMDS) by adding any new commands; use ISPDTLC to convert the BLG0EPSC source file, if necessary.
4. Test your modifications.
5. Move the modified BLGISPFE, BLGISPFA, and BLGISPFI panels to your production ISPPLIB data set.
6. If you changed BLG0CMDS, move it to your production ISPTLIB data set.

### Changing Keylists

ISPF provides the ability to use stored sets of function keys, called *keylists* on specific ISPF panels. Therefore, each ISPF panel can have a different keylist, thereby enabling you to provide context-specific function keys. You can change which keylist a particular panel uses, or you can change the contents of a keylist.
Changing Which Keylist a Panel Uses

To change which keylist a particular panel uses, change the name after KEYLIST in the )PANEL statement in the ISPF panel. To use the enhanced panel style without a keylist, edit the )PANEL statement and delete the KEYLIST parameter. Without specifying a keylist, the panel uses the profile function key settings.

Note: If you are using ISPF Version 4.2 or later, you can use DTL to change the keylist a specific panel uses. Change or remove the keylist attribute on the PANEL tag and then convert the source file using the DTL conversion utility.

Changing the Contents of a Keylist

If you are changing the command to be issued when the key is pressed, determine whether the command is an Information/Management command. If the command is not an Information/Management command, follow the procedure in 1 on page 5-19. If the command is an Information/Management command, you must first assign an alias to the command as described in “Assigning an Alias to an Information/Management Command.”

Note: If you are using ISPF Version 4.2 or later, you can change the Information/Management command table and keylists by using ISPF Dialog Tag Language. See “Using DTL to Change an Action Bar” on page 5-16 for more information.

Assigning an Alias to an Information/Management Command: To assign an alias to an Information/Management command, doing the following tasks:

1. Allocate both DDNAME ISPTABL and DDNAME ISPTLIB to the data set that contains the Information/Management BLG0KEYS and BLG0CMDS tables.

2. Select the COMMANDS option from the UTILITIES panel (3.9 if you are using the default ISPF panels). Enter BLG0 as the application ID. You see a table of Information/Management command aliases.

   Note: Do not change the BLG part of an entry. BLG associates the command with Information/Management.

3. Look for an existing alias to use. For example, to use the Information/Management PRINT command, locate:

   ALIAS BLG ;PRINT

   The alias name is BLGPRI. Use the alias name in the keylist KEYnDEF variable.

4. If the command you want to use does not already have an alias, add one to the table by doing the following:

   a. Insert a line into the command table.

   b. Create a new alias name (VERB). The name must be unique to the BLG0CMDS table, and the first 3 characters must be BLG.

   c. The ACTION must be of the following form:
ALIAS BLG ;your command and parms

Include the semicolon so that the command can be used when data is expected.

d. Use the END command to exit the COMMANDS option. You can now use the alias in your keylist.

5. Use the END command to exit the TABLES option.

Changing Function Keys on an ISPF Panel: To change the contents of the function keys that are displayed on a specific ISPF panel, do the following tasks:

1. Identify the KEYLIST name and APPLID by looking at the )PANEL KEYLIST statement in the appropriate panel. For example, for the enhanced panel style, BLGISPFE, the keylist name is BLGKEYLE and the application ID is BLG0.

2. Allocate both DDNAME ISPTABL and DDNAME ISPTLIB to the data set that contains the Information/Management BLG0KEYS and BLG0CMDS tables.

3. Use the DIALOG TEST option on the main ISPF panel, then the TABLES option (this is 7.4 if you use the default ISPF panels), find the row of the BLG0KEYS table that has KEYLISTN equal to the desired keylist name.

4. Type 3 to modify a row. To view the KEYLISTN value for the row, complete the panel as follows:
   
a. Type the name of the keylist in the Table name field.
   b. Type write in the Open table field.
   c. Specify a value in the By row number field.

5. Locate the key you want to change. Each key has 3 variables:
   
   **KEYnDEF** Specifies the command to process when this key is pressed.
   **KEYnLAB** Specifies the label to display at the bottom of the screen.
   **KEYnATR** Specifies when or whether to display the label. This field is case sensitive; be sure to use all uppercase when you type your choice of label display.

   - **LONG** Display the label only when all the keys are displayed.
   - **SHORT** Always display the label.
   - **NO** Never display the label.

6. Make your change and issue the END command to save it. The change you just made is effective for everyone who uses this table. If individual users want to change the function keys on a panel for their personal use, they can use the ISPF KEYLIST command.

Refer to the *ISPF Dialog Management Guide and Reference* for more information.
Adding Accelerator Keys and Separator Bars

You can add or change the accelerator keys and separator bars included in the panels for ISPF Version 4.2 or later users. See “Using DTL to Change an Action Bar” on page 5-16 for instructions on changing action bar definitions.

ISPF has certain rules and recommendations for accelerator keys that you must adhere to when you create or change the definitions:

- You cannot assign the following single keys as accelerator keys. They must be used in combination with another key or keys.
  - Ctrl
  - Shift
  - Alt
  - Alphabetic characters A through Z
  - Alphabetic characters a through z
  - Numeric characters 0 through 9

- Only one key in a combination can be a function key.

- You cannot assign the Shift key with any alphabetic or numeric character (A-Z, a-z, 0-9).

- If you only use two keys for a combination, one key must be from the following list. If you use three keys for a combination, two of the keys must be from this list.
  - Ctrl
  - Alt
  - Shift

  For either two- or three-key combinations, the other key must be one of these:
  - Insert
  - Delete
  - Backspace
  - Function keys F1 through F12
  - Alphabetic characters A through Z
  - Alphabetic characters a through z
  - Numeric characters 0 through 9

- You should avoid using the Alt key combined with only one character. (Use this combination for mnemonic access only.)

- You should avoid using a function key alone or Shift + a function key. (These combinations tend to cause confusion for users.)
Creating an Enhanced Panel Style Window

To create an enhanced panel style window, copy the BLGISPFE panel, or any other panel that was created using BLGISPFE as a model, to a new member in the appropriate ISPF panel data set. Give the new member the name that you want to use for selecting the panel. Then change the new member as described in “Changing the Code for an ISPF Panel” on page 5-8.

You can also add a keylist for a new panel. Refer to the ISPF Dialog Management Guide and Reference for details on how to do that.

Using the Standard Panel Style

To change the global default panel style to standard, copy BLGISPFD into BLGISPFM in the ISPPLIB concatenation.

An individual user can change his or her default to standard by issuing the following command:

WINDOW STANDARD,WINDOW SAVE

An individual user can temporarily switch to the standard style by issuing the following command:

WINDOW STANDARD

Refer to the User’s Guide for more information on the WINDOW command.
As the program administrator, you need database administration authority. This authority allows you to perform database administration tasks, which include DBCLEANUP and CHECK IN. It also allows you to create, update, and delete certain Information/Management records that have special functions like the ALIAS and data model records.

Data model records, which include validation, data view and data attribute records, are described in *Panel Modification Facility Guide*. The LOGSAVE record is described in the Data Propagation Tasks section in “Propagating Information/Management Data” on page 11-1. The ALIAS record is described in “Defining User Line Commands” on page 8-1. The COMMAND record is described in “Defining Command Aliases and Authorization” on page 9-1. The remainder of this chapter describes the DBCLEANUP and CHECK IN tasks.

If a system interrupt or ABEND occurs while a user is filing a record, the record file process may be unsuccessful. The record may be only partially filed, or it may be marked as being either busy or deleted from the database. You can restore these records using DBCLEANUP.

Applications running with the Information/Management application program interfaces can access records in the Information/Management database. It is possible for an application to check out a record from the database and fail to check it back in when processing is complete. You can use the CHECK IN option of database administration to check in these records to the database. You perform CHECK IN by requesting a list of checked out records and selecting from that list the records to be checked in. You can obtain a list of all the records checked out of a database or a list of all the records checked out to an application.
Performing **DBCLEANUP**

DBCLEANUP attempts to restore any partially filed records that are marked busy or deleted as a result of errors found in a previous file process. DBCLEANUP calls the BLG01336 program exit, which attempts to file any partially filed records. In addition, this program exit attempts to finish filing any records marked busy or partially deleted. If the file process on the busy or partially deleted records is not completed using **DBCLEANUP**, you can run the **BLGUT1** utility, which is documented in the *Operation and Maintenance Reference*.

From the Primary Options Menu for the System application, select **DBADMIN**. Type:

9

and press Enter.
The Database Administration Selection panel appears. Select **DBCLEANUP**.
Type:

1

and press Enter.

Any partially filed records are processed. A message indicates how many records were processed successfully and how many failed. To find out how to clean up any records that failed, refer to the *Diagnosis Guide*.

Type INIT and press Enter to return to the Primary Options Menu.
Performing **CHECK IN**

**CHECK IN** is used to check in to the database records that have been checked out by an application running with an application program interface. When an application running with an application program interface checks out a record, no other application can use that record. If this application fails to check in the record, you should then use **CHECK IN**.

With database administration authority you can obtain a list of checked out records and select records to be checked in from the list. A Checked Out Record list is compiled based on the data you enter on the Check In panel. You can request a list of all records checked out of a database, or you can ask for a list of all records checked out to an application ID. If you know that an application has not checked in a specific record, you can enter the record’s ID on the Check In panel. The Checked Out Record list is displayed, and by selecting the record from the list, you check it in.

On the Primary Options Menu for the System application, select **DBADMIN**.

Type:

9

and press Enter.
On BLG0Y100, the Database Administration Selection panel, select **CHECK IN**. Type:

2

and press Enter.
Performing CHECK IN

On BLG0Y200, the Check In panel, you must specify a database. The **Database** field is primed with the database number in your profile. If you press Enter after completing this field, the Checked Out Record list includes all records checked out of the specified database.

To obtain a list of records checked out to a specific application, put the database number in the **Database** field and type the application ID in the **Application ID** field.

To restrict the list to a specific record, put the database number in the **Database** field and type the record ID in the **Record ID** field.

---

**BLG0Y200**

**CHECK IN**

**DBADMIN**

Enter CHECK IN data; cursor placement or input line entry allowed.

1. Database.............<R> 5  
2. Application ID........... ________  
3. Record ID............ ________

To begin the function, press Enter without field modifications.

---
Performing CHECK IN

BLGITCRL, the Checked Out Record List panel, appears.

Select the record to be checked in. Move the cursor to the line command area next to the record ID that you want to check in. Type:

s

and press Enter.

<table>
<thead>
<tr>
<th>Record ID</th>
<th>APPL ID</th>
<th>Description</th>
<th>Abstract</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 00000835</td>
<td>APPL1</td>
<td>RESPONSE TIME TOO SLOW</td>
<td></td>
</tr>
<tr>
<td>2. 00000847</td>
<td>APPL2</td>
<td>PASSWORD DOESN'T WORK</td>
<td></td>
</tr>
<tr>
<td>3. 00000904</td>
<td>APPL2</td>
<td>NEEDS ANOTHER DISK DRIVE</td>
<td></td>
</tr>
<tr>
<td>4. 00000922</td>
<td>APPL3</td>
<td>UPGRADE CPU</td>
<td></td>
</tr>
<tr>
<td>5. 00000950</td>
<td>APPL1</td>
<td>REQUIRES GRAPHICS PRINTER</td>
<td></td>
</tr>
</tbody>
</table>

*** BOTTOM OF DATA ***

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

===>
Performing CHECK IN

A message that the record has been checked in appears.

To return to the Primary Options menu, type:

```
init
```

and press Enter.

```
BLGITCRL CHECKED OUT RECORD LIST LINE 4 OF 5

DATABASE: 5

<table>
<thead>
<tr>
<th>RECORD ID</th>
<th>APPL ID</th>
<th>DESCRIPTION</th>
<th>ABSTRACT</th>
</tr>
</thead>
<tbody>
<tr>
<td>4. 00000922</td>
<td>APPL3</td>
<td>UPGRADE CPU</td>
<td></td>
</tr>
<tr>
<td>5. 00000950</td>
<td>APPL1</td>
<td>REQUIRES GRAPHICS PRINTER</td>
<td></td>
</tr>
</tbody>
</table>

*** BOTTOM OF DATA ***

Line Cmds: S=Select
Type DOWN or UP to scroll the panel, or type END to exit the panel.
BLG031201 Record 00000922 was checked into the database successfully.

```
This chapter contains Product-sensitive Programming Interface and Associated Guidance Information.

This information is for the program administrator responsible for implementing Notification Management. You must understand Terminal Simulator Panels (TSPs), Terminal Simulator EXECs (TSXs), and the Panel Modification Facility (PMF) to perform the tasks described in this chapter. For information about TSPs and TSXs, refer to the *Terminal Simulator Guide and Reference*. For information about PMF, refer to the *Panel Modification Facility Guide*.

Notification Management is an Information/Management function that gives the people you designate a way to monitor the status of a problem, change, or activity record from its creation to its resolution. Designated people are notified by the system for certain conditions related to the record.

Notification Management encompasses two different functions: immediate notification and problem escalation.

Immediate notification sends a message to a specified user, or group of users, every time a problem, change, or activity record is entered or updated. Additionally, you can implement immediate notification for other types of records. The Integration Facility also provides immediate notification. Refer to *Integration Facility Guide* for more information.

Escalation processing handles only problem records. Problem escalation updates priorities and sends messages to people identified in rules records until a problem record is closed. Use problem escalation to alert people of unresolved problems in your data processing organization.

The two functions of Notification Management are separate and are started differently.

With Version 6.3 of Information/Management, enhancements were made to the Notification Management facility to:

- Use TSXs to perform notification functions. TSXs provide an easy way to construct complex mail messages and also allow for easier maintenance.
- Allow immediate notification mail and escalation messages to be sent using MVS TCP/IP Simple Mail Transfer Protocol (SMTP). Messages routed using SMTP can be processed by a locally installed electronic mail package such as Lotus Notes**.
- Allow messages to be sent directly from a user's address space, or added to a BLX-SP queue from which they will subsequently be sent. Placing the messages on a queue offloads notification processing from each user's address space.

- Enable immediate notification processing for records filed through API processing.

You can select which method to use to perform immediate notification—TSXs that use TCP/IP SMTP or TSPs that use TSO Send. The TSX method constructs a mail message using data from a filed record and sends the message using MVS TCP/IP SMTP. The TSP method builds a simple mail message using a filed record ID and sends the message to a TSO user using the TSO send command. You can disable immediate notification, or enable one of these two methods. Or, you can modify either of the sample methods provided to support processing available in your local environment.

See “Disabling Immediate Notification” on page 7-47 to disable immediate notification. See “Immediate Notification Using TSXs and TCP/IP” to enable TSXs and TCP/IP SMTP, or “Immediate Notification Using TSPs and TSO Send” on page 7-18 to enable TSPs and TSO send. For information on problem escalation, see “Understanding Problem Escalation” on page 7-49.

### Immediate Notification Using TSXs and TCP/IP

This section describes how to set up immediate notification using TSXs and TCP/IP SMTP. For details on how to enable Notification Management using TSPs and TSO Send, see “Immediate Notification Using TSPs and TSO Send” on page 7-18. That section describes the former traditional approach of setting up the Notification Management facility through the use of TSPs. Problem escalation is explained in “Understanding Problem Escalation” on page 7-49. For a discussion of immediate notification with the Integration Facility, refer to Integration Facility Guide.

### Notification Processing with TSXs

Information/Management provides sample TSXs to perform immediate notification. TSXs can provide an easier way to construct a complex mail message than using traditional TSPs. The TSXs provided can:

- Build the notification message when a record is filed.
- Send the message using TCP/IP SMTP processing or add the message to a BLX-SP queue.
- Extract the message from the BLX-SP queue and send the message using TCP/IP SMTP mail.

Information/Management provides sample TSXs that can be invoked when problem, change, and activity records are filed. You can invoke a TSX anywhere in the panel dialogs to initiate notification processing. The sample TSXs provide a way to send a detailed mail message to the assignee of a record when it is filed.
You can modify the TSXs to include different data in the message, send the message to someone other than the assignee, or use a different mail mechanism other than TCP/IP SMTP. You can also write your own TSX to notify users at times other than when filing records. For instance, you could write a TSX that would kick off notification processing when the user makes a selection.

By using TSXs, Information/Management can easily construct the mail message using data from the filed record and provide a sample method of sending the mail using MVS TCP/IP SMTP.

An overview of the processing done by TSXs for TCP/IP SMTP mail follows:

- When a record requiring notification is filed, a TSX is invoked.
- The TSX extracts the assignee to notify and other information from the record.
- If an assignee is found, the TSX looks in the USERS record to find any mail addresses defined for that person. If no address is found, a default address is used.
  
  A user exit, BLGUSERS, accesses the address information in the USERS record.
  
  The mail message is built.
  
  The TSX appends TCP/IP SMTP header and trailer information to the message to put it in a format that can be processed through SMTP.
  
  The TSX calls a user exit, BLGNSYAL, to allocate a SYSOUT data set with a destination of SMTP.
  
  Once the data set is allocated, a TSX writes the message data to it. User exit BLGNSYFR frees the data set, which serves to route the message to the SMTP server. No TSO services are required, so API-generated records and records generated through interactive use can use this method of immediate notification.
  
  Alternatively, the TSX can send the message to a BLX-SP queue for later processing. (The default is not to queue the messages.)
  
  TCP/IP SMTP routes the mail using your installation's mail configuration. Once messages reach this point, Information/Management no longer has control of the message.

**Immediate Notification with TSXs**

As shipped, immediate notification is performed by a set of TSPs that are called when a problem, change, or activity record is filed. The TSPs use the **Assignee Name** field in the record to determine to whom to send the notification message. If the TSPs cannot determine to whom to send the notification message, the TSPs send it to a default user ID that is specified in the TSPs. You must enable or disable immediate notification before allowing your users to create or update problem, change, or activity records. If you want to use the TSXs that are provided to handle immediate notification processing, you must also change the panels that call the TSPs to call the TSXs instead. The following section describes how to
enable immediate notification so that it uses TSXs and TCP/IP SMTP. If you prefer to disable immediate notification, see “Disabling Immediate Notification” on page 7-47 for complete instructions.

**Enabling Immediate Notification with TSXs Using TCP/IP SMTP**

Before you can use immediate notification, you must perform the following steps to enable it. The TSXs supplied with Information/Management reside in the SBLMTSX data set. You can use any MVS editor to update them. You should copy any TSX you want to modify to another data set before making modifications.

To run a TSX, you must have the BLGTSX DD allocated as described in the “Starting Information/Management” section of the Planning and Installation Guide and Reference.

**Note:** Integration Facility users: If you plan to use TSXs, you can generally follow steps 1 through 7, but be sure to use the TSXs defined for the Integration Facility (BTNXCN07 and BTNXP03) as described in Integration Facility Guide instead. The examples shown in this section show the use of the BLGTXNOT TSX. You will need to make comparable changes in the Integration Facility TSXs.

1. Change the default mail addresses in the TSX. The sample TSX you would change for immediate notification is BLGTXNOT. Refer to “Changing the Default Mail Address” on page 7-5 for instructions.

2. Decide if you want to enable immediate or queued processing. With immediate processing, the mail message is constructed and sent when a record is filed, all within the user's address space. With queued processing, the mail message is constructed when the record is filed, but instead of being sent immediately, it is added to one of three BLX-SP queues and later extracted from the queue and sent. Immediate processing is the default; no action is required to implement it. To implement queued processing, see “Queued Processing” on page 7-6 for more information.

3. Decide if you want to use TCP/IP for message routing. If you do, define the correct TCP/IP SMTP mail header and trailer information for your environment in the BLGTXNOT TSX. See “Defining TCP/IP SMTP Header and Trailer Information” on page 7-7 for more information. Also, you should ensure that TCP/IP is installed and that SMTP is configured to be enabled. (For more information on SMTP, refer to the IBM TCP/IP V3R1 for MVS Customization and Administration Guide, SC31-7134.) If you do not want to use TCP/IP SMTP, modify the TSX logic to support the mail processing of your choice.

4. Define the logical record length, SYSOUT class, destination, and ddname of the data set dynamically allocated to TCP/IP SMTP to match your local environment. See “Defining the SMTP Data Set” on page 7-8 for more information.

5. Modify the ‘FROM:’ mail line defined in the BLGTXNOT TSX buildmsg subroutine to include the mail address to use as the from address. See “Modifying the "From" Mail Line” on page 7-8 for more information.
6. Determine the format of the mail message to send and what record data to include in the notification message. This data could be different for each type of record. For instance, you might want to include priority data from problem records but not change records.

7. Using the Panel Modification Facility, change the following control panels to invoke the appropriate TSXs.
   
   - BLG1A111 Problem records
   - BLG1A121 Change records
   - BLG1A131 Activity records

   To determine which TSX to invoke, refer to “TSP to TSX Equivalent” on page 7-10 for a list of immediate notification TSXs.

8. If you haven't already done so, create a USERS record. See “Creating the USERS Record” on page 7-10 for more information.

9. Perform a simple test to ensure that when a record is filed, it is routed to the proper mail address. If you did not set up the USERS record, the message should be routed to the default address.

changing the default mail address

The default mail address in the TSXs as shipped by IBM must be changed to enable notification messages to be routed to the program administrator (or some other designated person) if an assignee or alias is not found in the USERS record. The default mail address must be changed for each type of record that the notification TSXs process. You can also change the processing, if desired; for example, you might choose not to send a message if the filed record does not contain an assignee.

For immediate notification, change the default mail addresses in the BLGTXNOT TSX, which can be found in the SBLMTSX data set and changed using an MVS editor. The data set(s) that contain the notification TSXs must be allocated to the BLGTSX DD to invoke TSXs for immediate notification. (For information on allocating the BLGTSX DD, refer to the Planning and Installation Guide and Reference.)

The BLGTXNOT TSX sends the notification message for problem, change, and activity records. Determine the default address to use for each type of record, and modify this TSX to define them. The default address is used when the record does not have an assignee or when the assignee does not have any mail addresses defined in the USERS record.

Refer to Figure 5 on page 7-6 for an example of where to change the default mail addresses in the BLGTXNOT TSX. The figure shows an extract from the sample BLGTXNOT TSX. You would, for instance, enter a default address for problem records on the DEFAULT_ADDRESS_PROBLEM line. Your own TSX or other similar TSXs (such as those used with the Integration Facility or Escalation processing) may contain variations.
Immediate Notification Using TSXs and TCP/IP

Figure 5. Portion of the BLGTXNOT TSX Shipped with Information/Management

Queued Processing
Immediate processing, where a mail message is constructed and sent when a record is filed, is the default type of mail processing that is specified in the BLGTXNOT TSX shipped with Information/Management. If you would like to reduce the time a user must wait wait for a record to be filed, you can defer some of the mail processing to a later time without directly impacting the user. BLGTXNOT can build a mail message and add it to a BLX-SP queue. You can alter this logic to minimize the processing performed when the record is filed. For example, you could place just the record ID of the record being filed onto the BLX-SP queue. When the record ID is later pulled off the queue, the mail message could be built and sent at that time.

You can define up to three BLX-SP queues to contain mail messages. You can specify the queue names and limits on the number of items that can be placed onto each queue. For instructions on how to set the MAILQ parameter of the BLX-SP...
parameters member to specify names of mail queues and queue limits, refer to the *Planning and Installation Guide and Reference*. To query the BLX-SP to find out how many mail items exist on the queues or to modify the queue limits, refer to the *Operation and Maintenance Reference* for a description of the MAILQ operator command.

To use queued processing, you must modify the BLGTXNOT TSX to specify QUEUE as the **notification_type** (an example of BLGTXNOT is shown in Figure 5 on page 7-6). You must also use one or more TSXs to extract the mail from the queue(s) and send the mail. These TSXs can be run using Information/Management in batch or interactively. Sample TSX BLGTXDEQ shows how to dequeue and send mail using TCP/IP SMTP.

When the shutdown command is issued for the BLX-SP, the system checks to see if there are any mail items on any of the queues. If there are, a message is issued indicating that the shutdown will halt until the queues are empty or the wait time is exceeded.

Use of a message queue will impact the processing of the BLX-SP in two ways:

- The BLX-SP will have more work to do because it will place items on and off the mail queues.
- At shutdown, if the MAILQWAITTM parameter is specified in the session-parameters member, processing is suspended until either all of the mail queues are empty or the wait time has been exceeded. If you do not specify MAILQWAITTM, shutdown occurs immediately regardless of the status of the mail queues.

You can, however, specify how long the BLX-SP will wait upon shutdown for mail to clear off of the queues. The wait time is broken into 10 second intervals. If the queues are not cleared by the time you specify in the MAILQWAITTM parameter in the BLXPRM startup, the queues are flushed. The mail items are lost and shutdown processing continues.

Therefore, if you need to stop the BLX-SP, you may want to wait until the mail is processed. Once the shutdown starts, there is no way to query the number of items on the queues, change the wait time, or stop the shutdown.

Refer to the *Planning and Installation Guide and Reference* for more information on the BLX-SP parameters member and use of the MAILQWAITTM parameter.

**Defining TCP/IP SMTP Header and Trailer Information**

As part of the overall setup to enable the Notification Management facility to route messages using MVS TCP/IP SMTP, you must define the correct SMTP mail header and trailer information for your environment.

The header and trailer information is defined in the BLGTXNOT TSX. (For an example, see the code starting at the HDR.1 line in Figure 5 on page 7-6.) After the BLGTXNOT TSX looks in the USERS record for mail addresses defined to the assignee, it builds the mail message and includes this SMTP header and trailer information. Without this data, the mail messages will not be routed successfully.
For more information on SMTP, refer to the *IBM TCP/IP V3R1 for MVS* Customization and Administration Guide, SC31-7134.

**Defining the SMTP Data Set**

As part of the immediate notification processing performed by BLGTXNOT, a SYSOUT data set is dynamically allocated to the destination of SMTP. The BLGTXNOT TSX writes the mail message to this data set, which sends the mail to the SMTP server, and frees the data set.

You must ensure that the BLGTXNOT TSX contains the correct data set attributes and destination information for the SYSOUT data set. The logical record length, SYSOUT class, and destination (TCP/IP SMTP node and userid) of the data set dynamically allocated to SMTP should match values required by your local environment.

Refer to Figure 5 on page 7-6 for an example of where these definitions occur in the sample BLGTXNOT TSX.

**Modifying the "From" Mail Line**

The BLGTXNOT TSX must be modified to contain an address indicating where mail messages are coming from. This information is included in the mail message.

Refer to Figure 6 on page 7-9 for an example of where you would change the FROM mail address in the BLGTXNOT TSX (see `call addmsg('From: mailaddress');`). The FROM mail address can refer to your Information/Management system.
Immediate Notification Using TSXs and TCP/IP

Defining the SMTP Data Set for Use with Dequeue Processing
Information/Management provides a TSX, BLGTXDEQ, which shows how to dequeue mail messages from the BLX-SP queue, if message queuing is selected rather than processing in the user's address space. If you elect to queue mail messages, you must also modify the BLGTXDEQ TSX or the TSX(s) you are using to dequeue and process mail to define the SMTP data set attributes and destination. The definition should include the logical record length, SYSOUT class, destination, and ddname of the SMTP data set for your environment.

Additionally, you must modify the DEQMAIL parameters in the BLGTXDEQ TSX to match the queue name used in your environment.

Changing the Message Format
As part of the setup procedure to enable immediate notification with TSXs and TCP/IP SMTP, you must determine the format of the mail message to send, and what data to include from the record in the notification message. The contents could be different for each type of record, and additional data can be added.

To determine what to include in the mail message, first review the sample BLGTXNOT TSX provided with Information/Management to become familiar with its function and the message content that would normally be provided (see Figure 6). For example, it contains message data such as the record ID, reporter,
Immediate Notification Using TSXs and TCP/IP

date, assignee, status, description, and priority. Then, decide what alterations are required for your environment and make the necessary changes in your TSX.

**TSP to TSX Equivalent**

As part of your setup procedure to enable notification processing with TSXs, you need to modify panels to invoke the correct TSX. Use the following tables to determine what changes to make. For example, modify the BLG1A111 panel to invoke the BLGTSXPE TSX for problem creation notification, rather than the BLGTSPPE TSX.

### Table 6. Information/Management Immediate Notification TSPs and TSX Equivalents

<table>
<thead>
<tr>
<th>TSP name</th>
<th>TSX name</th>
<th>Purpose</th>
<th>Invoked by</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGTSPPE</td>
<td>BLGTSXPE</td>
<td>Problem create notification</td>
<td>BLG1A111</td>
</tr>
<tr>
<td>BLGTSPPU</td>
<td>BLGTSXPU</td>
<td>Problem update notification</td>
<td>BLG1A111</td>
</tr>
<tr>
<td>BLGTSPCE</td>
<td>BLGTSXCE</td>
<td>Change create notification</td>
<td>BLG1A121</td>
</tr>
<tr>
<td>BLGTSPCU</td>
<td>BLGTSXCU</td>
<td>Change update notification</td>
<td>BLG1A121</td>
</tr>
<tr>
<td>BLGTSPAE</td>
<td>BLGTSXAE</td>
<td>Activity create notification</td>
<td>BLG1A131</td>
</tr>
<tr>
<td>BLGTSPAU</td>
<td>BLGTSXAU</td>
<td>Activity update notification</td>
<td>BLG1A131</td>
</tr>
</tbody>
</table>

A separate TSX is still called for each record type and record action, but the TSX calls the BLGTXNOT TSX passing the type of record and the record action, as shown in the next table.

Table 7 summarizes other immediate notification TSXs that are shipped with Information/Management.

### Table 7. Other Significant Information/Management Immediate Notification TSXs

<table>
<thead>
<tr>
<th>TSX name</th>
<th>Purpose</th>
<th>Processing</th>
<th>Invoked by</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGTXNOT</td>
<td>Perform notification for all record types.</td>
<td>Read record just filed and extract information for mail message. Send message through TCP/IP SMTP or queue message to BLX-SP for later processing.</td>
<td>TSXs BLGTSXPE, BLGTSXPU, BLGTSXCE, BLGTSXCU, BLGTSXAE, BLGTSXAU</td>
</tr>
<tr>
<td>BLGTXDEQ</td>
<td>Sample TSX that shows dequeuing mail built using BLGTXNOT.</td>
<td>Dequeues the mail and sends it using TCP/IP SMTP.</td>
<td>n/a</td>
</tr>
</tbody>
</table>

### Creating the USERS Record

Before you can use immediate notification effectively, you must create a USERS record. Having a USERS record prevents a default user ID from receiving all notification messages. You can have only one USERS record.

The USERS record you create must contain:
- The names (individual users or groups of users, such as departments) that are possible assignees for a problem, change, or activity record.

You can also specify a distribution list which exists on another mail system, such as Lotus Notes.

A USERS record can contain up to 19,274 names of assignees or aliases. (Aliases are typically used for Escalation message notification, or with the Integration Facility, where you might have a list of people such as problem reviewers or change committee members who would need to be notified of record changes.)

- The mail address associated with each name (this could be an MVS userid, or an electronic mail address such as a Lotus Notes ID or Internet address). If you choose to use TCP/IP SMTP, this address must be capable of being processed by TCP/IP SMTP.

Immediate notification takes the assignee name information from the record being filed, locates it in the USERS record, and sends a notification message to the corresponding mail address or addresses. If it is unable to determine to whom to send the notification message, it sends the message to the default user ID that you specified in the TSX.

The sample immediate notification TSXs assume that the mail addresses are contained in the USERS record. You can choose to use some other method to store the mail addresses and modify the TSX logic accordingly. Depending on your environment, you may be able to make the assignee name itself a mail address in the target mail system, thus eliminating the address look-up step.

User exit BLGUSERS can be called from a TSX to extract mail addresses from the USERS record. (See 7-83 for information about BLGUSERS.)
To create the USERS record, on the Primary Options Menu for the System application, select **Entry**. Type 5 and press Enter.

```
BLGOEN10 --- PRIMARY OPTIONS MENU --- APPLICATION: SYSTEM

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. DBADMIN...........Perform database administration.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997

```
The USERS record is reference information. From this panel, select **Reference**.
Type 2 and press Enter.
To create the USERS record, select **Users**. Type 1 and press Enter.

```plaintext
+ 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.

+--------------------------------- SELECT ITEM ----------------------------------+

===> 1
```
The USERS record is assigned a record ID of USERS.

The standard Information/Management USERS TO NOTIFY panel appears. Since many electronic addresses may be too long to fit in the ID/Class column and since user exit BLGUSERS cannot extract addresses entered into the ID/Class fields, rather than use this standard panel, type in option 61 and press Enter to get the list processor panel for mail information entry.

Enter the assignee/alias names and addresses of the individuals or groups whom you want to receive immediate notification messages. The assignee names can be 1 to 15 alphanumeric positions and can contain #, @, $, &; and / characters. (You can also enter a privilege class as the assignee name, but you must repeat the entry and type in the address of each person in the privilege class.) The mail address can be 1 to 60 characters.

After you enter the data, type end and press Enter to return to the USERS TO NOTIFY panel.
**Immediate Notification Using TSXs and TCP/IP**

**BLGUSER**  
**MAIL INFORMATION ENTRY**  
**LINE 1 OF 10**

USE....Enter names and addresses for use by Notification and/or Escalation.  
**RECORD: USERS**

<table>
<thead>
<tr>
<th>Assignee/ Alias</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>JohnSmith</td>
<td><a href="mailto:johnsmith@sample.com">johnsmith@sample.com</a></td>
</tr>
<tr>
<td>SarahWood</td>
<td><a href="mailto:srwood@sample.com">srwood@sample.com</a></td>
</tr>
<tr>
<td>SallyWhite</td>
<td>swhite</td>
</tr>
<tr>
<td>Helpdesk</td>
<td>Operations1</td>
</tr>
<tr>
<td>Helpdesk</td>
<td><a href="mailto:johnsmith@sample.com">johnsmith@sample.com</a></td>
</tr>
</tbody>
</table>

Line Cmns: 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
Since the **Address** field can contain 60 characters, only part of the field is shown on the panel. If your data is longer than what you can type on this panel, you can type an **L2** (for Address) in the line command area to the left of the row that you want to update, and press Enter to go to the assisted-entry panel where you can enter the complete address. Or, you can use the RIGHT and LEFT scroll commands to scroll the **Address** column to enter more data.

Type **end** again on the **USERS TO NOTIFY** panel to file your entries.

---

<table>
<thead>
<tr>
<th>BLG08901</th>
<th>USERS TO NOTIFY</th>
<th>1 OF 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enter assignee and ID/class data; cursor placement or input line entry allowed.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Assignee Name</strong></td>
<td><strong>ID/Class</strong></td>
</tr>
<tr>
<td>1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>34.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>36.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>39.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>41.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>42.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>43.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>44.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>46.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>47.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>49.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>50.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>52.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>53.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>55.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>58.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>59.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>60.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**61. Enter list of assignees and addresses.**

**Press Enter to add more entries.**

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

***=> end***
The task of creating a USERS record is finished. A message tells you that the record was stored.

As more people become targets of notification, you must add their names and mail addresses to the USERS record.

--- PRIMARY OPTIONS MENU --- APPLICATION: SYSTEM

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. DBADMIN........Perform database administration.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997

Because you can have only one USERS record, this record cannot be copied and it does not appear on inquiry panel flows. The record can, however, be displayed, updated, printed, or deleted. If you attempt to create another USERS record, you are taken into update mode for the existing USERS record.

Note: The BLGLUSER panel is intended for use with e-mail notification and the use of TSXs. It is not recommended that you mix assignees on the BLG0B901-3 panel(s) with assignees on the BLGLUSER panel. You should use one or the other method of entering the assignee names. If you decide to use TSXs, you should delete the USERS record you previously created using TSPs (if you had one), and create it again using the BLGLUSER panel. To make entry of the new record easier, you should issue the PRINT R USERS command to print the old USERS record first before deleting it.

Immediate Notification Using TSPs and TSO Send

This section describes how to enable immediate notification using the standard TSP approach. For information on how to set up immediate notification using TSXs, refer to “Immediate Notification Using TSXs and TCP/IP” on page 7-2. You can enable only one of these approaches. If you are using TSXs and TCP/IP SMTP, ignore this section.
The method of immediate notification described in this section is one that is controlled by a set of TSPs that are called when a problem, change, or activity record is filed. The TSPs use the Assignee Name field in the record to determine to whom to send the notification message. If the TSPs cannot determine to whom to send the notification message, the TSPs send it to a default user ID that is specified in the TSPs. You must enable or disable immediate notification before allowing your users to create or update problem, change, or activity records. The following describes how to enable the standard TSP method of doing immediate notification. If you prefer to disable immediate notification, see “Disabling Immediate Notification” on page 7-47 for complete instructions.

**Enabling Immediate Notification**

Before you can use immediate notification, you must perform the following steps to enable the function:

1. Delete the BRANCH statement from the TSPs marked for immediate notification in the following lists.
2. Change the default user ID in the same TSPs.
3. Create a USERS record.

Depending on the needs of your organization, you may also want to alter some of the TSPs for immediate notification. To enable immediate notification, change the TSPs indicated by an asterisk (*) in the following list.

- The TSPs used for immediate notification in Problem Management are:
  - `BLGTSPPE*` Send notification that a problem record was entered.
  - `BLGNOTPE` Called by `BLGTSPPE` when the ID/Class associated with the Assignee name field in the problem record is a privilege class.
  - `BLGTSPPU*` Send notification that a problem record was updated.
  - `BLGNOTPU` Called by `BLGTSPPU` when the ID/Class associated with the Assignee name field in the problem record is a privilege class.

- The TSPs used for immediate notification in Change Management are:
  - `BLGTSPCE*` Send notification that a change record was entered.
  - `BLGNOTCE` Called by `BLGTSPCE` when the ID/Class associated with the Assignee name field in the change record is a privilege class.
  - `BLGTSPCU*` Send notification that a change record was updated.
  - `BLGNOTCU` Called by `BLGTSPCU` when the ID/Class associated with the Assignee name field in the change record is a privilege class.

- The TSPs used for immediate notification of activity records are:
Immediate Notification Using TSPs and TSO Send

BLGTSPAEB* Send notification that an activity record was entered.

BLGNOTAE Called by BLGTSPAEB when the ID/Class associated with the Assignee name field in the activity record is a privilege class.

BLGTSPAU* Send notification that an activity record was updated.

BLGNOTAU Called by BLGTSPAU when the ID/Class associated with the Assignee name field in the activity record is a privilege class.

Deleting the BRANCH Statements
Using PMF, delete the BRANCH control line that branches to the LABEL control line called MESSAGE in the following TSPs:

For these records: Change these TSPs:
Problem records BLGTSPPE and BLGTSPPU
Change records BLGTSPCE and BLGTSPCU
Activity records BLGTSPAE and BLGTSPAU.

For example, this panel shows the first 14 lines of BLGTSPPE. The fifth line is the BRANCH control line you delete.

<table>
<thead>
<tr>
<th>Line</th>
<th>Cmds</th>
<th>Control Line Summary</th>
<th>Line 1 of 106</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>LABEL BLGTSPPE NOTIFY THE APPROPRIATE PERSON</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>USEREXIT BLGTSAPI TEST FOR API ENVIRONMENT NO NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>TESTFIELD LEAVE TSP 0 NO NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BRANCH MESSAGE</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>USEREXIT CALL BLGESGCB TO GET STORAGE NO NO</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LABEL 00000000 CHANGE 'DEFAULT' TO THE ID 00000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LABEL 00000000 OF YOUR PROBLEM ADMINISTRATOR 00</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LABEL 000000 IN LINE 12. SPACE FILL IF 00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LABEL 0000 NEEDED. LEAVE POSITIONS 00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LABEL 0000 1-8 BLANK. POSITION 17 00000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>LABEL 0000 MUST CONTAIN A '.' 0000000000000000</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>MOVEVAR DEFAULT</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>USEREXIT CALL BLGESPUV TO SAVE 'DEFAULT' YES NO</td>
<td></td>
</tr>
</tbody>
</table>

Line Cmds: D=Delete U=Update
Type DOWN, UP, LEFT, or RIGHT to scroll the panel, or type END to exit EXIT

==>
Changing the Default User ID

In the same TSPs you just changed, you must also change the default user ID. The default user ID in the TSPs as shipped by IBM is DEFAULT (see the MOVEVAR control line in the previous screen). You must change DEFAULT to the MVS user ID of the program administrator who is responsible for notification management. A comment in each of these TSPs helps you identify where to make the changes.

The notification message is sent to the default ID if any one of these conditions is true:

- The database has no USERS record.
- The problem, change, or activity record has no assignee name.
- The assignee name in the problem, change, or activity record is not in the USERS record.
- The assignee name is found in the USERS record, but the ID/Class field for that name is blank.
- The assignee name identifies a privilege class record, but the user filing the problem, change, or activity record does not have privilege class display authority.

When the ID/Class field for the specified assignee name is found in the USERS record, it is identified either as a privilege class or as an individual.

- If the ID/Class is a privilege class record, a message is sent to each of the first 20 users listed in the privilege class record. (TSO SEND restricts you to sending messages to only 20 users.)
- If the ID/Class field is not a privilege class record, it is assumed to be an individual, and the notification message is sent to that user ID.

Note: If an assignee name is listed more than once in the USERS record, the notification message is sent to the ID/Class field associated with the last occurrence of that name in the USERS record.
Creating the USERS Record

Before you can use immediate notification effectively, you must create a USERS record. Having a USERS record prevents a default user ID from receiving all notification messages. You can have only one USERS record in the database.

The privilege class is used when the assignee name represents a group. If you use privilege classes to secure your data, the people creating and updating problem, change, or activity records must have display authority for privilege class records.

The USERS record you create must contain:

- The names (individual users or groups of users, such as departments) that are possible assignees for a problem, change, or activity record. A USERS record can contain up to 90 names.
- The MVS user ID or privilege class associated with each name.

Immediate notification takes the assignee name information from the record being filed, locates it in the USERS record, and sends a notification message to the corresponding MVS user ID, or to the first 20 eligible users in the privilege class. If it is unable to determine to whom to send the notification message, it sends the message to the default user ID that you specified in the TSP.

On the Primary Options Menu for the System application, select **Entry**. Type 5 and press Enter.

--- PRIMARY OPTIONS MENU ---
APPLICATION: SYSTEM

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. DBADMIN........Perform database administration.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997

--- 5 ---
The USERS record is assigned a record ID of USERS.

The standard Information/Management USERS TO NOTIFY panel appears. Since many electronic addresses may be too long to fit in the ID/Class column and since user exit BLGUSERS cannot extract addresses entered into the ID/Class fields, rather than use this standard panel, type in option 61 and press Enter to get the list processor panel for mail information entry.

The USERS record is reference information. From this panel, select Reference. 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.

------------------------- SELECT ITEM ----------------------------+

==>> 2
```
To create the USERS record, select **Users**. Type 1 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.|
+------------------------ SELECT ITEM -----------------------------+

===> 1
```
The USERS record is assigned a record ID of USERS.

Enter the assignee names and user IDs or privilege classes of all individuals or groups whom you want to receive immediate notification messages.

You can include 30 names on this panel. If you want to make additional entries, press Enter after completing this panel, and a second entry panel appears. A third panel appears if you press Enter again. The USERS record can contain a maximum of 90 names as shipped.

<table>
<thead>
<tr>
<th>BLG08901</th>
<th>USERS TO NOTIFY</th>
<th>1 OF 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter assignee and ID/class data; cursor placement or input line entry allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignee Name</td>
<td>ID/Class</td>
<td>Assignee Name</td>
</tr>
<tr>
<td>5.</td>
<td>6.</td>
<td>35.</td>
</tr>
<tr>
<td>7.</td>
<td>8.</td>
<td>37.</td>
</tr>
<tr>
<td>11.</td>
<td>12.</td>
<td>41.</td>
</tr>
<tr>
<td>13.</td>
<td>14.</td>
<td>43.</td>
</tr>
<tr>
<td>15.</td>
<td>16.</td>
<td>45.</td>
</tr>
<tr>
<td>17.</td>
<td>18.</td>
<td>47.</td>
</tr>
<tr>
<td>19.</td>
<td>20.</td>
<td>49.</td>
</tr>
<tr>
<td>21.</td>
<td>22.</td>
<td>51.</td>
</tr>
<tr>
<td>23.</td>
<td>24.</td>
<td>53.</td>
</tr>
<tr>
<td>25.</td>
<td>26.</td>
<td>55.</td>
</tr>
<tr>
<td>27.</td>
<td>28.</td>
<td>57.</td>
</tr>
<tr>
<td>29.</td>
<td>30.</td>
<td>59.</td>
</tr>
</tbody>
</table>

61. Enter list of assignees and addresses. Press Enter to add more entries. When you finish, type END to save or CANCEL to discard any changes.

Note: Option 61 is used for entering data for the TSX method of immediate notification using TCP/IP SMTP. For instructions on using option 61 and TSXs, refer instead to “Creating the USERS Record” on page 7-10 for information.

After you enter all of the assignee names and ID/Classes for your organization, type end and press Enter to file the record.
The task of creating a USERS record is finished. A message tells you that the record was stored.

Expanding the USERS Record

Your organization may need more than the 90 names you can list in the USERS record. You can expand the USERS record to include additional names.

Assume you need to add 60 additional names. To do this, you must create a new, unique s-word for every field you add. Because each name includes two fields, Assignee name and ID/Class, you must add a total of 120 new s-words to the dictionary to add 60 new names to a USERS record. Refer to the Panel Modification Facility Guide for complete instructions on adding new s-words to the dictionary.

You must modify 3 panels to allow these additional 60 names. Those panels are:

- BLG0B903
- BLG0L903
- BLG1A197.

You must create 4 new panels to allow these additional 60 names:

1. Copy panel BLG0B903 twice.
a. Name the first copy BLG0B904.
b. Name the second copy BLG0B905.

2. Copy panel BLG0L903 twice.
   a. Name the first copy BLG0L904.
   b. Name the second copy BLG0L905.

Change the fields on the panels as indicated in Table 8.

<table>
<thead>
<tr>
<th>On Panel:</th>
<th>Null Target Field</th>
<th>Change From:</th>
<th>Change to:</th>
<th>Function Code Index Field</th>
<th>Change From:</th>
<th>Change to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0B903</td>
<td>BLG1A911</td>
<td>BLG0B904</td>
<td></td>
<td>(no change)</td>
<td>(No change)</td>
<td>(no change)</td>
</tr>
<tr>
<td>BLG0B904</td>
<td>BLG1A911</td>
<td>BLG0B905</td>
<td></td>
<td>(no change)</td>
<td>(No change)</td>
<td>(no change)</td>
</tr>
<tr>
<td>BLG0B905</td>
<td>BLG1A911</td>
<td>(no change)</td>
<td>(no change)</td>
<td>(no change)</td>
<td>(no change)</td>
<td>(no change)</td>
</tr>
<tr>
<td>BLG0L903</td>
<td>(blank)</td>
<td>BLG0L904</td>
<td>000D</td>
<td>000A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0L904</td>
<td>(blank)</td>
<td>BLG0L905</td>
<td>000D</td>
<td>000A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0L905</td>
<td>(blank)</td>
<td>(no change)</td>
<td>000D</td>
<td>(no change)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For each of the 4 new panels you created, select **Summary** on panel BLM8CU70. This allows you to update each field and change the s-words. Every field on panels BLG0B904 and BLG0B905 must have a unique s-word. Use these same s-words on panels BLG0L904 and BLG0L905. Each panel has 60 fields, one for the **ID/Class** field and one for the **Assignee name**.

You must also update control panel BLG1A197 and insert 120 new ADD control lines. For each new control line, update the **Structured word index** field with one of the new s-words you defined in the dictionary. Also, update the **Prefix index** field as follows:

- Assignee name control lines: 056B
- ID/Class control lines: 056A

When you finish creating these control lines, file the panel. You must delete the current USERS record so that the updated control panel (BLG1A197) will initialize the new fields.

Test your changes. If your changes do not work, verify that the s-words and prefix indexes are correct in the BLG1A197 control panel. Also check to be sure that the **Null target** fields on the BLG0B903, BLG0B904, and BLG0B905 appear as shown in Table 8.
**Changing the Notification Message**

The TSPs and the notification text in each follows:

<table>
<thead>
<tr>
<th>TSP Name</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGNOTAE</td>
<td>Activity record–rnid created</td>
</tr>
<tr>
<td>BLGNOTAU</td>
<td>Activity record–rnid updated</td>
</tr>
<tr>
<td>BLGTSPAE</td>
<td>Activity record–rnid created</td>
</tr>
<tr>
<td>BLGTSPAU</td>
<td>Activity record–rnid updated</td>
</tr>
<tr>
<td>BLGNOTCE</td>
<td>Change record–rnid created</td>
</tr>
<tr>
<td>BLGNOTCU</td>
<td>Change record–rnid updated</td>
</tr>
<tr>
<td>BLGTSPCE</td>
<td>Change record–rnid created</td>
</tr>
<tr>
<td>BLGTSPCU</td>
<td>Change record–rnid updated</td>
</tr>
<tr>
<td>BLGNOTPE</td>
<td>Problem record–rnid created</td>
</tr>
<tr>
<td>BLGNOTPU</td>
<td>Problem record–rnid updated</td>
</tr>
<tr>
<td>BLGTSPPE</td>
<td>Problem record–rnid created</td>
</tr>
<tr>
<td>BLGTSPPU</td>
<td>Problem record–rnid updated</td>
</tr>
</tbody>
</table>

Three control panels support immediate notification:

- BLG1A111 — Problem record
- BLG1A121 — Change record
- BLG1A131 — Activity record.

These control panels call the required TSP when a record is filed. You need not change the control panels if you change either the text of the notification message or the message recipient.

You build the message text with a series of MOVEVAR control lines. You can change the message text in the TSPs by modifying these control lines.

The TSPs that generate the messages follow the naming convention BLGTSP.xy and BLGNOT.xy, where x is P for problem records, C for change records, and A for activity records. The y character is E for newly entered records or U for updated records.

This example uses TSP BLGTSPPE, for newly entered problem records, to demonstrate how to modify the notification message to include the **Description** field. This example works only when the assignee name is not specified, the assignee name is not in the USERS record (the default ID is used), or the assignee name is an individual. If you wish to include the **Description** field in the notification message when the assignee name is a privilege class, TSP BLGNOTPE must be modified as TSP BLGTSPPE is modified in this example.
Select **PMF** from the Management application Primary Options Menu (BLG0EN20). Then select **Panel update** from the Panel Modification Facility options panel (BLM8C000). The Panel Name Entry panel shown here appears.

In the **Panel name** field, type `blgtsppe` and press Enter twice.
Immediate Notification Using TSPs and TSO Send

On the command line, type 1 and press Enter.

```
+ BLM8CU90 ------- TERMINAL SIMULATOR PANEL UPDATE ---------- PMF+*

OPTIONS:

  1. ABSTRACT....Modify description of this panel.
  2. COMMON.....Modify common panel control information.
  4. SUMMARY.....Display summary of control information.
  5. TEST........Process panel in test mode.
  6. FILE.......Panel update is complete, store panel.

+-------------------------- SELECT OPTION --------------------------+

===>
```

Panel BLM8CU91 appears. On the command line, type control and press Enter.

```
BLM8CU91  TERMINAL SIMULATOR PANEL UPDATE  EXTERNALS

+------------------------------------------------------------------+
| BLGTSPPPE  PMF |
+------------------------------------------------------------------+

This TSP runs when a problem record is filed. It sends a notification message to users for new problem records.

Modify textual data within the box. To modify control data, type CONTROL on the command line. When you finish, type END to save or CANCEL to discard any changes.

```control

The control lines for the TSPs appear. This TSP is large and does a lot of processing, but the actual notification message is generated with only 17 control
Immediate Notification Using TSPs and TSO Send

These control lines start at line 68. If you have deleted the BRANCH statement to enable immediate notification, these control lines start at line 67.

**Note:** The starting line might be different if you have made other modifications to this TSP or if you are using an earlier version of Information/Management.

The 17 lines begin with a LABEL command. In this case, the LABEL command is a comment line. The comment is BUILD AND SEND THE MESSAGE. The 8 control lines that follow this LABEL line are those that build the notification message. Two of those 8 lines must not be changed. They are the USEREXIT commands with an 8 in the **Literal data** field. **Do not** modify these lines; these are the lines that move the record ID and user ID into the message. In this example, line 70 and line 72 must not be changed.

These are control lines that create notification messages. (The next three BLMITUCU panel illustrations do not show all the data that appears on the panel. They also assume you have deleted the BRANCH statement to enable immediate notification.)

<table>
<thead>
<tr>
<th>BLMITUCU FUNCTION LINE SUMMARY</th>
<th>LINE 67 OF 104</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTION NAME</td>
<td>LABEL NAME</td>
</tr>
<tr>
<td>67. LABEL</td>
<td>SENDMSG</td>
</tr>
<tr>
<td>68. MOVEVAR</td>
<td>SELECT CMD(SEND '</td>
</tr>
<tr>
<td>69. MOVEVAR</td>
<td>'PROBLEM RECORD-</td>
</tr>
<tr>
<td>70. USEREXIT</td>
<td>8</td>
</tr>
<tr>
<td>71. MOVEVAR</td>
<td>CREATED''' USER(</td>
</tr>
<tr>
<td>72. USEREXIT</td>
<td>8</td>
</tr>
<tr>
<td>73. USEREXIT</td>
<td>CALL BLGESFCB TO FREE STORAGE NO NO</td>
</tr>
<tr>
<td>74. MOVEVAR</td>
<td>) LOGON )</td>
</tr>
<tr>
<td>75. ISPEXEC</td>
<td></td>
</tr>
</tbody>
</table>

The notification message is built by moving all the parts of the message into the variable data area and then processing the data in the variable data area as a TSO SEND command with the ISPEXEC command line.

The first MOVEVAR command (line 68) that was created has the **Replace Data** field set to YES to clear the variable data area and to insert the new data. Any additional MOVEVAR commands you add must have the **Replace Data** field set to NO. The data that is associated with the additional MOVEVAR command lines is concatenated to the data in the first MOVEVAR command line.
The actual text that appears on the screen as the notification message is the data between the two sets of triple quotation marks. Any data you add must be between these sets of quotation marks. Any data you add will be in the case in which you enter it. To prevent interfering with the USEREXITs and other parts of the message, the description data is concatenated at the end of the message. That is, the description field follows the word CREATED in line 71.

Line 71 must be broken into two separate control lines, and the additional commands you add must be inserted between them. Split the original MOVEVAR control line at the closing set of triple quotation marks. In the line command area of line 71 use the I (INSERT) command.

---

**BLMITUCU**

**FUNCTION LINE SUMMARY**

**LINE 67 OF 104**

<table>
<thead>
<tr>
<th>FUNCTION NAME</th>
<th>LABEL</th>
<th>LITERAL</th>
<th>GET</th>
<th>APPLY</th>
<th>FIELD</th>
<th>VAR</th>
<th>NOT</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>67. LABEL SENDMSG</td>
<td>BUILD AND SEND THE MESSAGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68. MOVEVAR</td>
<td>SELECT CMD(SEND '</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69. MOVEVAR</td>
<td>'PROBLEM RECORD-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70. USEREXIT</td>
<td>8</td>
<td>YES NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71. MOVEVAR</td>
<td>CREATED'' USER(</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72. USEREXIT</td>
<td>8</td>
<td>YES NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73. USEREXIT</td>
<td>CALL BLGESF CB TO FREE STORAGE</td>
<td>NO NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74. MOVEVAR</td>
<td>) LOGON )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75. ISPEXEC</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Line Cmds:** A=After  C=Copy  D=Delete  I=Insert  M=Move  R=Repeat  U=Update  
Type DOWN or UP to scroll the panel, or type END to exit.

---
Create a MOVEVAR control line.

```plaintext
+ BLM6FUNC ---------------- FUNCTION NAME ---------------- NO PREFIX+

USE....Enter name of function to execute for this control line.

NOTE...Any commands issued here must be preceded by a ';'.

| ADDDATA | LABEL | SETFIELD |
| BRANCH  | LINK  | TESTFIELD |
| CLEAR   | MESSAGE | TESTFLOW  |
| FINDSDATA | MOVEVAR | TRACE   |
| FINDSJRNL | PRINT | UNFLATTEN |
| FLATTEN | PROCESS | USEREXIT |
| ISPEXEC | RETURN | WORDFIX |

+--------------------- REPLY AS DEFINED ---------------------------+

====> movevar
```
In the **Literal data** field, type:

```plaintext
'\'' USER()
```

**Note:** Be aware that the data will be kept in the case in which you enter the data.
The panels which follow reflect the term *USER* rather than *user*.

On the command line, type `end` and press Enter to leave the panel.

```
BLMBCU90  MOVEVAR SPECIFICATION  PANEL: BLGTSPPE

Enter 'MOVEVAR' control data; cursor placement or input line entry allowed.

1. TSCA Field Name.................
2. Literal Data.................... `'' USER()
3. Replace data.................... NO_
4. Data conversion................ NO_

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

```
Now, remove that information from the original MOVEVAR control line. In the line command area of line 71 use the U (UPDATE) command.

```
BLMITUCU FUNCTION LINE SUMMARY   LINE 67 OF 105

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>LABEL</th>
<th>LITERAL</th>
<th>GET</th>
<th>APPLY</th>
<th>FIELD</th>
<th>NAME</th>
<th>NAME</th>
<th>DATA</th>
<th>VAR</th>
<th>NOT</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>67. LABEL</td>
<td>SENDMSG</td>
<td>BUILD AND SEND THE MESSAGE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68. MOVEVAR</td>
<td>SELECT CMD(SEND '</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69. MOVEVAR</td>
<td>'PROBLEM RECORD-</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70. USEREXIT</td>
<td>8 YES NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71. MOVEVAR</td>
<td>CREATED USER(</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72. MOVEVAR</td>
<td>'' USER(</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73. USEREXIT</td>
<td>8 YES NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74. USEREXIT</td>
<td>CALL BLGESFCB TO FREE STORAGE NO NO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75. MOVEVAR</td>
<td>) LOGON )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76. ISPEXEC</td>
<td>YES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Line Cmds: A=After C=Copy D=Delete I=Insert M=Move R=Repeat U=Update Type DOWN or UP to scroll the panel, or type END to exit.
```
In the **Literal data** field, blank out the data after the word CREATED, and add a dash as shown. On the command line, type `end` and press enter to leave the panel.
To add the **Description** field data to the notification message, in the line command area of line 71, use the I (INSERT) command to add two new control lines.

```
67. LABEL SENDMSG BUILD AND SEND THE MESSAGE
68. MOVEVAR SELECT CMD(SEND ')
69. MOVEVAR 'PROBLEM RECORD-
70. USEREXIT 8 YES NO
71. MOVEVAR CREATED -
72. MOVEVAR ''' USER(
73. USEREXIT 8 YES NO
74. USEREXIT CALL BLGESFCB TO FREE STORAGE NO NO
75. MOVEVAR ) LOGON )
76. ISPEXEC YES
```

The first control line to create is a MOVEVAR control line. This line places a blank in the variable data area. Be sure the **Replace data** field is set to NO.

On the command line, type `end` and press Enter to leave this panel.
The second control line to create is a MOVEVAR control line. This line takes the **Description** field from the TSCA field TSCASDF and concatenates it to the end of the data in the variable data area.

In the **TSCA field name** field, type:

```
tscasdf
```

Be sure the **Replace data** field is set to NO.

On the command line, type `end` and press Enter to leave this panel.

---

BLM8CU9O MOVEVAR SPECIFICATION PANEL: BLGTSPPE

Enter 'MOVEVAR' control data; cursor placement or input line entry allowed.

```
1. TSCA Field Name............ tscasdf_
2. Literal Data................ ________________________________
3. Replace data............... NO_
4. Data conversion............. NO_

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

```
The revised TSP looks like this.

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>LABEL</th>
<th>LITERAL</th>
<th>GET APPLY</th>
<th>FIELD</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>NAME</td>
<td>DATA</td>
<td>VAR</td>
<td>NOT</td>
</tr>
<tr>
<td>67.</td>
<td>LABEL</td>
<td>SENDMSG</td>
<td>BUILD AND SEND THE MESSAGE</td>
<td></td>
</tr>
<tr>
<td>68.</td>
<td>MOVEVAR</td>
<td>SELECT CMD(SEND '</td>
<td></td>
<td></td>
</tr>
<tr>
<td>69.</td>
<td>MOVEVAR</td>
<td>'PROBLEM RECORD-</td>
<td></td>
<td></td>
</tr>
<tr>
<td>70.</td>
<td>USEREXIT</td>
<td>8</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>71.</td>
<td>MOVEVAR</td>
<td>CREATED -</td>
<td></td>
<td></td>
</tr>
<tr>
<td>72.</td>
<td>MOVEVAR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73.</td>
<td>MOVEVAR</td>
<td>'' USER(</td>
<td>TSCASDF</td>
<td></td>
</tr>
<tr>
<td>74.</td>
<td>MOVEVAR</td>
<td>8</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>75.</td>
<td>USEREXIT</td>
<td>CALL BLGESFCB TO FREE STORAGE</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>76.</td>
<td>USEREXIT</td>
<td>) LOGON</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Line Cmds: A=After C=Copy D=Delete I=Insert M=Move R=Repeat U=Update Type DOWN or UP to scroll the panel, or type END to exit.

Now, we must insert several other control lines to load the TSCA field TSCASDF with the Description from the problem record.

In the line command area of line 67, use the I (INSERT) command to add six new control lines.
The first control line to create is a MOVEVAR control line.

In the **TSCA field name** field, type:

tscalfid

Be sure the **Replace data** field is set to YES.

On the command line, type `end` and press Enter to leave this panel.
The second control line to create is an ADDDATA control line.

In the **Literal data** field, type:

DI R

Be sure the **Get variable data** field is set to YES.

On the command line, type end and press Enter to leave this panel.

```
BLM8CU9A ADDDATA SPECIFICATION PANEL: BLGTSPPE

Enter 'ADDDATA' control data; cursor placement or input line entry allowed.

1. Structured word index.... 0000 Structured word.. __________
2. Get variable data........ YES Word acronym..... __________
3. Literal data............. DI R

WHEN COMPLETE REPLY END (TO SAVE) OR CANCEL (NO SAVE)

```
The third control line to create is a **PROCESS** control line. This line will cause the last record filed to be displayed.

In the **Error label name** field, type **BADREAD**.

On the command line, type `end` and press Enter to leave this panel.

```
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Error label name.....&lt;R&gt;</td>
<td>BADREAD</td>
</tr>
<tr>
<td>2. Save existing messages?..</td>
<td>NO_</td>
</tr>
</tbody>
</table>
```

WHEN COMPLETE REPLY END (TO SAVE) OR CANCEL (NO SAVE)
The fourth control line to create is a FINDSDATA control line. The s-word index associated with the Description field is S0E0F.

In the Structured word index field, type Oe0f and in the Word occurrence field, type last.

On the command line, type end and press Enter to leave this panel.

---

Note: It is important to keep the Description field a required field in problem, change, and activity entry and update panels, so that the newly added FINDSDATA command always finds data to add to the notification message.
The fifth control line to create is an ADDDATA control line.

In the **Literal data** field, type ;CA. Be sure the **Get variable data** field is set to **NO**.

On the command line, type `end` and press Enter to leave this panel.
The sixth control line to create is a PROCESS control line.

In the **Error label name** field, type:

badread

On the command line, type `end` and press Enter to leave this panel.
The final revised TSP now looks like this.

<table>
<thead>
<tr>
<th>FUNCTION LINE SUMMARY</th>
<th>LINE 67 OF 107</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTION NAME</td>
<td>LABEL</td>
</tr>
<tr>
<td>NAME</td>
<td>NAME</td>
</tr>
<tr>
<td>67.</td>
<td>LABEL</td>
</tr>
<tr>
<td>68.</td>
<td>MOVEVAR</td>
</tr>
<tr>
<td>69.</td>
<td>ADDDATA</td>
</tr>
<tr>
<td>70.</td>
<td>PROCESS</td>
</tr>
<tr>
<td>71.</td>
<td>FINDSDATA</td>
</tr>
<tr>
<td>72.</td>
<td>ADDDATA</td>
</tr>
<tr>
<td>73.</td>
<td>PROCESS</td>
</tr>
<tr>
<td>74.</td>
<td>MOVEVAR</td>
</tr>
<tr>
<td>75.</td>
<td>MOVEVAR</td>
</tr>
<tr>
<td>76.</td>
<td>USEREXIT</td>
</tr>
<tr>
<td>77.</td>
<td>MOVEVAR</td>
</tr>
<tr>
<td>78.</td>
<td>MOVEVAR</td>
</tr>
<tr>
<td>80.</td>
<td>MOVEVAR</td>
</tr>
<tr>
<td>81.</td>
<td>USEREXIT</td>
</tr>
<tr>
<td>82.</td>
<td>USEREXIT</td>
</tr>
<tr>
<td>83.</td>
<td>MOVEVAR</td>
</tr>
</tbody>
</table>

Line Cmds: A=After C=Copy D=Delete I=Insert M=Move R=Repeat U=Update Type DOWN or UP to scroll the panel, or type END to exit.

Assuming that your user ID is DON and that your ID is the default, you can use this IRC on the Primary Options Menu to build a problem record after you modify BLGTSPPE:

5,1,1,=,14,open,25,THIS IS A TEST OF THE NOTIFICATION MESSAGE,,9

Because you do not specify an assignee name and because DON is not the name of a privilege class, the message goes to your default user ID, DON. The resulting notification message is:

'PROBLEM RECORD-000000051 CREATED - THIS IS A TEST OF THE NOTIFICATION MESSAGE' DON

See “Notification Management User Exits” on page 7-65 for complete descriptions of the user exits called by the immediate notification TSPs. Refer to the Terminal Simulator Guide and Reference for complete information on creating control lines in a TSP.
Disabling Immediate Notification

You can disable immediate notification for all records if you do not want to use it. If you want to use immediate notification for only one type of record, but not for the others, you can do that, too.

Update the control panels by removing the two sets of multiple test groups that end with the 001B function code so that immediate notification messages are no longer sent. The control panels that require updating follow:

- Problem records: BLG1A111
- Change records: BLG1A121
- Activity records: BLG1A131

The following example shows how you can disable notification for Problem Management. Disabling Change and Activity notification follows a similar sequence. To begin, select PMF from the Management application Primary Options Menu; then select Panel update from the Panel Modification Facility options panel.

To update the control panel for Problem Management, in the Panel name field, type:

```
blg1a111
```

and press Enter twice.
Select **Abstract** to display details about the control panel. Type 1 and press Enter.

On the command line of the Control Panel Update abstract description panel, type `control` and press Enter.

Modify textual data within the box. To modify control data, type `CONTROL` on the command line. When you finish, type `END` to save or `CANCEL` to discard any changes.
On the Function Line Summary panel, use the D (DELETE) line command to delete the two sets of multiple test groups that end with the 001B function code. Each test group includes a TEST and FLOW function type. Type dd as shown and press Enter.

If you had enabled immediate notification using TSXs and TCP/IP SMTP, the names BLGTSPPE and BLGTSPPU in the Function Line Summary panel would be changed to BLGTSXPE and BLGTSXPU.

File your changes, and copy the panel into a read panel data set.

### Understanding Problem Escalation

You can use the problem escalation facility of Information/Management to inform interested users of a problem’s status and its priority until the problem is closed. Escalation processing runs in batch mode and at the intervals you set. The escalation of problem records is controlled by a list of criteria that you define in rules records, a record type defined specifically for problem escalation.

The escalation facility uses the rules records criteria to create a search argument that identifies problem records that you want escalated. For each problem that is identified, Information/Management sends an alert message to one or more user IDs or mail addresses, as specified in the rules record. The **Current priority** field and the **Escalation level** field of the problem record are updated each time a message is sent.

Escalation messages can be processed using MVS TCP/IP SMTP mail. To take advantage of this function, you must build the escalation message using the existing *ddname* or *dsname* processing as described later in this section.
Implementing Problem Escalation

Before you can implement escalation processing, you need to understand the process and perform the following tasks:

1. Read this section for an overview of the problem escalation process and the implementation tasks.
2. Establish your installation’s escalation criteria within the problem management process. If you want to use escalation criteria that are different from those shipped by IBM, use PMF to update the rules record panels.
3. Define the rules records for each set of criteria that you want to monitor.
4. Create the JCL to run the escalation batch job and set the interval at which this job is to be run. Some sample JCL, called BLMESCAL, is provided for you in the SBLMSAMP data set.

Understanding the Escalation Process

The escalation facility uses RULES records to create a search argument to identify problem records to escalate. For each problem that is escalated, an alert message is sent to one or more user IDs specified in the RULES record. You can also use each RULES record user ID field as an “alias” that is defined along with one or more mail addresses in the USERS record. The alias can be a privilege class name or a name of your choice to identify one or more people that should be sent an escalation message. If you use this method, the escalation message is sent to each address defined for the alias. Current Priority and Escalation Level fields in the problem record are also updated.

The following steps outline the processing performed by TSP BLGESCAL. See “Notification Management User Exits” on page 7-65 for information about the escalation control block (ESCB) and the user exits for notification management.

1. Search for all RULES records and obtain a search results list. For complete information about creating RULES records, see “Creating Rules Records” on page 7-53.
2. Select each RULES record, one at a time from the search results list, and do the following:
   a. Move selected information from the RULES record to the ESCB.
   b. Move the search criteria from the RULES record to the ESCB.
   c. Store the escalation command template in the ESCB. The template contains the command and an outline of the format used to send the escalation alert message.
   d. Build a search argument to find all nonclosed problem records in the database.

---

1 A nonclosed problem is a problem with any status other than closed.
e. Determine which of the escalation criteria exist in the RULES record. Add them to the search argument. For example, if Key item affected is one of the search criteria and the RULES record contains an entry of REPORTS in the Key item affected field, then COMK/REPORTS is added to the search argument (COMK is the p-word for Key item affected). If the Key item affected field in the RULES record is blank, this p-word is not added to the search argument.

f. Issue a SEARCH command to get a list of all nonclosed problem records that are candidates for escalation according to this RULES record.

g. Process each problem record:

1) Subtract the Date and Time fields in the problem record from the current date and time. The result is the calculated duration. The escalation level in the problem record determines whether the Date entered and Time entered fields or the Date last altered and Time last altered fields are used for the subtraction operation.

2) Compare the calculated duration against the duration in the RULES record to determine whether to issue an alert. The escalation level in the problem record determines whether the level 1, 2, or 3 duration should be used.

3) Build the alert message, using either the data in the variable data area for the Terminal Simulator Communications Area (TSCA) or a specified message panel.

4) Substitute the alert message and the appropriate information from the ESCB in the escalation command template.

5) Send the alert message.

6) If the Priority adjustment field in the RULES record is not 0, update the Current priority field in the problem record.

7) Update the Escalation level in the problem record.

8) File the record.

Note: When problem records are updated because of escalation processing, immediate notification messages are not sent.

9) Processing is complete for one problem record. Repeat steps 2.g.1 through 2.g.8 until all problem records in the search results list are processed.

h. Processing is complete for one RULES record. Repeat steps 2.a through 2.g until all RULES records in the search results list created in step 1 have been processed.

3. Processing is complete. Print the messages on the chain and exit from Information/Management.
The Escalation TSPs and TSXs

The BLGESCAL TSP controls the escalation facility. This TSP calls other TSPs and TSXs to perform different escalation functions. The following table names each TSP and its purpose. The processing steps in this table refer to “Understanding the Escalation Process” on page 7-50.

<table>
<thead>
<tr>
<th>TSP Name</th>
<th>Purpose of the TSP</th>
<th>Processing Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGESCAL</td>
<td>Escalation facility control.</td>
<td>1, 2.h, 3</td>
</tr>
<tr>
<td>BLGESC01</td>
<td>Finds the escalation criteria in a RULES record and stores the criteria in the ESCB.</td>
<td>2.a, 2.b, 2.d, 2.e, 2.f</td>
</tr>
<tr>
<td>BLGESC02</td>
<td>Controls the problem record loop area.</td>
<td>2.g.8, 2.g.9</td>
</tr>
<tr>
<td>BLGESC03</td>
<td>Gets data from the RULES record and stores it in the ESCB.</td>
<td>2.a</td>
</tr>
<tr>
<td>BLGESC04</td>
<td>(Continuation of BLGESC03.) Also stores the notification command template in the ESCB.</td>
<td>2.a, 2.c</td>
</tr>
<tr>
<td>BLGESC05</td>
<td>Determines the escalation level of a problem record.</td>
<td>N/A</td>
</tr>
<tr>
<td>BLGESC06</td>
<td>Performs processing for a problem record based on the escalation level.</td>
<td>2.g.1 through 2.g.7</td>
</tr>
<tr>
<td>BLGESC07</td>
<td>Builds a notification message.</td>
<td>2.g.3</td>
</tr>
</tbody>
</table>

In addition to these TSPs, a TSX is available for escalation processing:

<table>
<thead>
<tr>
<th>TSX Name</th>
<th>Purpose of the TSX</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGESCX6</td>
<td>Sends the escalation message via TCP/IP SMTP</td>
</tr>
</tbody>
</table>

If you elect to use the BLGESCX6 TSX, you must modify some escalation TSPs. See “Using the BLGESCX6 TSX with TCP/IP SMTP” on page 7-64 for more information.
Creating Rules Records

Familiarize yourself with the escalation criteria defined in the escalation facility and decide whether they fit your installation’s needs. You must create your rules records before escalation processing can be done. The security of your rules records is controlled by the privilege class records in your database. The Privilege Class Record panels include authority for rules records.

The criteria used in the escalation facility as shipped from IBM uses the following fields in the rules records to find matching fields in problem records:

- **Initial priority**
- **Problem type**
- **Location code**
- **Key item affected**.

Create a rules record for each set of unique search criteria that you want escalated. Refer to the *User’s Guide* for information about creating records.

To establish your site’s escalation criteria, make a list of the possible combinations of escalation criteria that are valid for your site. For example, perhaps your location has the following:

- Four initial priority values: 01, 02, 03, and 04
- Two problem types: Hardware and Software
- One location: A10
- Two key items affected: TAPE01 and DISK03B

Because every problem reported has a location code of A10, you need not use the **Location code** field as part of your search argument. Therefore, this field is not an escalation criterion for your installation.

Continuing with this example, you come up with a list of 16 rules records that include every possible situation that can occur. Table 10 lists these rules records.

<table>
<thead>
<tr>
<th>Rules Record Number</th>
<th>Initial Priority</th>
<th>Key Item Affected</th>
<th>Problem Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01</td>
<td>TAPE01</td>
<td>Hardware</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>TAPE01</td>
<td>Software</td>
</tr>
<tr>
<td>3</td>
<td>01</td>
<td>DISK03B</td>
<td>Hardware</td>
</tr>
<tr>
<td>4</td>
<td>01</td>
<td>DISK03B</td>
<td>Software</td>
</tr>
<tr>
<td>5</td>
<td>02</td>
<td>TAPE01</td>
<td>Hardware</td>
</tr>
<tr>
<td>6</td>
<td>02</td>
<td>TAPE01</td>
<td>Software</td>
</tr>
<tr>
<td>7</td>
<td>02</td>
<td>DISK03B</td>
<td>Hardware</td>
</tr>
<tr>
<td>8</td>
<td>02</td>
<td>DISK03B</td>
<td>Software</td>
</tr>
<tr>
<td>9</td>
<td>03</td>
<td>TAPE01</td>
<td>Hardware</td>
</tr>
</tbody>
</table>
Table 10 (Page 2 of 2). Example Rules Records

<table>
<thead>
<tr>
<th>Rules Record Number</th>
<th>Initial Priority</th>
<th>Key Item Affected</th>
<th>Problem Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>03</td>
<td>TAPE01</td>
<td>Software</td>
</tr>
<tr>
<td>11</td>
<td>03</td>
<td>DISK03B</td>
<td>Hardware</td>
</tr>
<tr>
<td>12</td>
<td>03</td>
<td>DISK03B</td>
<td>Software</td>
</tr>
<tr>
<td>13</td>
<td>04</td>
<td>TAPE01</td>
<td>Hardware</td>
</tr>
<tr>
<td>14</td>
<td>04</td>
<td>TAPE01</td>
<td>Software</td>
</tr>
<tr>
<td>15</td>
<td>04</td>
<td>DISK03B</td>
<td>Hardware</td>
</tr>
<tr>
<td>16</td>
<td>04</td>
<td>DISK03B</td>
<td>Software</td>
</tr>
</tbody>
</table>

Scanning this list, you find that some combinations cannot logically occur in your installation. For example, you never open software problems with an initial priority of 01. Therefore, you do not need to create rules records numbered 2 and 4.

Suppose you have assigned only one person to monitor all problems with the key item TAPE01. You, therefore, need to create only one rules record for the key item affected TAPE01, leaving the Initial priority and Problem type fields blank. In this example, you do not need to create rules records numbered 1, 2, 5, 6, 9, 10, 13, and 14. The one rules record that you need to create uses the Key item affected field as the only escalation criterion.
You can add, delete, or modify rules records at any time. A rules record that has no escalation criteria results in all nonclosed problem records being candidates for escalation. Rules records are Management application records, so be sure to start on Primary Options Menu, BLGOEN20, for the Management application. To get to the Escalation Rules Entry panel, type:

5,4

and press enter.
Understanding Problem Escalation

The Escalation Rules Entry panel contains several required fields. Some of these fields are primed. The **Current priority** field is primed with a value of 01, and the **Inc/Dec priority** field is primed with a minus sign (-). You can modify both fields.

When the criteria in this rules record cause a problem record to be escalated, the current priority in the problem record is adjusted based on the values in these fields. The values shown cause the **Priority adjust** field in the problem record to be decremented (because of the minus sign) by 1.

The Level 1, 2, and 3 IDs can be the actual userid on the Level 1, 2, and 3 nodes or they can be aliases as defined in the USERS record. If they are aliases, do not use the node fields.

**Understanding the Escalation Search Argument**

The escalation facility initially searches for all rules records. It then builds a search argument for all nonclosed problems that match the criteria found in the first rules record escalation criteria fields. For example, if the **Initial priority** field in your first rules record has a value of 03, the search argument finds all nonclosed problems with an initial priority of 03. If the **Problem type** field also contains the entry SOFTWARE, the search argument finds all nonclosed problems with an initial priority of 03 and a problem type of software. This logic is continued for the **Location code** field. When the escalation function finishes processing the problem records that require escalation based on the first rules record, it continues to the next rules record and processes the appropriate problem records based on the next rules record criteria. This process continues until all of the rules records are used.
Table 11 on page 7-57 illustrates how a search argument is built.

**Note:** The SUSPEND command is issued first so that when you are finished processing the problem records, you return to the next rules record in the list.

<table>
<thead>
<tr>
<th>Initial Priority</th>
<th>Problem Type</th>
<th>Location Code</th>
<th>Key Item Affected</th>
<th>Search Argument</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>SUS,6,1,SE + ~STAC/CLOSED PRII/3</td>
</tr>
<tr>
<td>3</td>
<td>software</td>
<td></td>
<td></td>
<td>SUS,6,1,SE + ~STAC/CLOSED PRII/3 TYPE/SOFTWARE</td>
</tr>
<tr>
<td>3</td>
<td>software</td>
<td>A10</td>
<td></td>
<td>SUS,6,1,SE + ~STAC/CLOSED PRII/3 TYPE/SOFTWARE LOCC/A10</td>
</tr>
<tr>
<td>3</td>
<td>software</td>
<td>A10</td>
<td>DISK03B</td>
<td>SUS,6,1,SE + ~STAC/CLOSED PRII/3 TYPE/SOFTWARE LOCC/A10 COMK/DISK03B</td>
</tr>
</tbody>
</table>

All problem records have an *Escalation level* field that is not seen by the user. An escalation level of 1 is added to a problem record the first time the record is filed. The *Escalation level* field is incremented by 1 each time escalation occurs for this record (up to a maximum value of 4) to enable the escalation processor to determine which levels of escalation have been performed.

If you try to escalate a problem record that was created in your database on some version before 4.1 and the record has not been subsequently updated under 4.1 or a later version, the *Escalation level* field is added to the record the first time the record is updated.

The BLGESCAL TSP processes the rules records one at a time, locating all of the nonclosed problem records that match the criteria set in each rules record. Then BLGESCAL determines which records to escalate. The contents of the *Escalation level* field in each problem record determines which levels of duration, node, user ID, date set, and time set to use in the processing. If a record that meets the criteria does not contain an escalation level, an escalation level of 1 is assumed.

Next, the BLGESCAL TSP sends the escalation message, updates the *Current priority* field in the problem record according to the values in the rules record, and increments the number of the escalation level in the problem record. The maximum escalation level is 4. When this level is reached, it remains at 4 until the problem record is closed.

If the duration value in the rules record for the current escalation level is blank, the *Escalation level* field is incremented; however, the priority remains unchanged, and no message is sent.
Escalating Problems

Table 12 illustrates how two different rules records cause problem records to be escalated. These records illustrate how the escalation level and how long the problem has been active determine whether a problem is escalated and whether an alert message is sent.

<table>
<thead>
<tr>
<th>Rules Record Field</th>
<th>Rules Record 1</th>
<th>Rules Record 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority adjust</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Inc/Dec Priority</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Initial priority</td>
<td>20</td>
<td>01</td>
</tr>
<tr>
<td>Problem type</td>
<td>SWBATCH</td>
<td></td>
</tr>
<tr>
<td>Location code</td>
<td></td>
<td>DALLAS</td>
</tr>
<tr>
<td>Key item affected</td>
<td>ANYITEM</td>
<td>ANYITEM</td>
</tr>
<tr>
<td>Level 1 duration</td>
<td>00:01:00</td>
<td>00:12:00</td>
</tr>
<tr>
<td>Level 1 ID to notify</td>
<td>USER01</td>
<td>USER01</td>
</tr>
<tr>
<td>Level 1 node</td>
<td>ABC01</td>
<td>DEF01</td>
</tr>
<tr>
<td>Level 2 duration</td>
<td>00:06:00</td>
<td></td>
</tr>
<tr>
<td>Level 2 ID to notify</td>
<td>USER02</td>
<td></td>
</tr>
<tr>
<td>Level 2 node</td>
<td>ABC01</td>
<td></td>
</tr>
<tr>
<td>Level 3 duration</td>
<td>07:00:00</td>
<td>01:00:00</td>
</tr>
<tr>
<td>Level 3 ID to notify</td>
<td>USER03</td>
<td>USER03</td>
</tr>
<tr>
<td>Level 3 node</td>
<td>ABC02</td>
<td>DEF01</td>
</tr>
</tbody>
</table>

Rules Record 1

In this example:

- When a problem with a type of SWBATCH, an initial priority of 20, and a key item affected of ANYITEM has existed for at least one hour and has not been closed, USER01 at ABC01 is sent an alert message, and the Current priority field in the problem record is reduced by 1. The escalation level in the record is raised to 2.

- When the same problem has existed for at least 6 hours and has not been closed, USER01 and USER02 at ABC01 are sent a message, and the Current priority field in the problem record is reduced by 1. The escalation level in the record is raised to 3.
When the same problem has existed for at least 7 days and has not been closed, USER01 and USER02 at ABC01 and USER03 at ABC02 are sent a message, and the Current priority field in the problem record is reduced by 1. The escalation level in the record is raised to 4.

Alert messages are sent to all three users and the Current priority field is lowered by 1 every 7 days after that (until it reaches 1, where it stops) until the problem record is closed. If the problem record is updated during this time, alert messages are sent every 7 days from when the record was last updated and filed. The escalation level in the problem record, once it reaches 4, does not change.

**Rules Record 2**

In this example:

- When a problem with a location of DALLAS and an initial priority of 01 has existed for at least 12 hours and has not been closed, USER01 at DEF01 is sent an alert message, and the Current priority field in the problem record is raised by 1. The escalation level in the record is raised to 2.

- The next time the escalation batch job runs, Information/Management looks at the Level 2 duration field in the rules record and finds a blank. This tells the system that no escalation message needs to be sent, and the current priority is not raised. The escalation level in the problem record is raised to 3.

- The system checks the Level 3 duration field on succeeding cycles, and, when the problem has existed for one day, a message is sent to USER03 at DEF01, as well as to USER01 at DEF01. The Current priority field in the record is increased by 1, and the escalation level is raised to 4.

- Assuming that no other updates are made to the problem record, alert messages are sent to both USER01 and USER03, and the Current priority field in the problem record is incremented by 1 every day after that until the problem record is closed. If the record is updated, alert messages are sent every day afterward, based on when the record was last updated and filed.

**Note:** When a record is updated, the escalation clock restarts. For example, if the duration at escalation level 3 is 5 hours and the record is updated 4 hours after the last message was sent, the next message is sent when 9 hours have passed. If the record is not updated again, another message is sent in 5 hours.

This example assumes you are not using the BLGESCX6 TSX to route escalation messages using TCP/IP SMTP.
Sending the Alert Message

Table 13 shows which users are notified at each escalation level.

<table>
<thead>
<tr>
<th>Escalation Level</th>
<th>Duration Used</th>
<th>Date/Time Used</th>
<th>Messages Sent to</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Level 1 duration</td>
<td>Date/time entered</td>
<td>Level 1 user ID or address</td>
</tr>
<tr>
<td>2</td>
<td>Level 2 duration</td>
<td>Date/time entered</td>
<td>Level 1 and 2 user IDs or addresses</td>
</tr>
<tr>
<td>3</td>
<td>Level 3 duration</td>
<td>Date/Time entered</td>
<td>Level 1, 2, and 3 user IDs or addresses</td>
</tr>
<tr>
<td>4</td>
<td>Level 3 duration</td>
<td>Date/Time last altered</td>
<td>Level 1, 2, and 3 user IDs or addresses</td>
</tr>
</tbody>
</table>

**Note:** If you want only one user notified in any rules record, specify that user as a level 3 user, leave levels 1 and 2 blank and complete only the Level 3 duration, Level 3 ID to notify, and Level 3 node fields. You only need Level 3 node if your Level 3 ID to notify is a userid. If it is an alias, you do not need to specify anything in the Level 3 node. If the level 3 fields are blank, the escalation messages are sent only twice. After the level 1 and level 2 user IDs are notified, no other messages are sent.

Starting Escalation Processing

Problem escalation is designed to be started through a batch job that is submitted on a periodic basis to MVS. This batch job starts the BLGESCAL TSP that reads all the RULES records. If you have five different RULES records, the TSP performs five different searches against the nonclosed problem records in the database.

Creating the JCL for the Escalation Batch Job

BLMESCAL in the sample library (SBLMSAMP) contains JCL you can use to start the escalation batch job. Modify BLMESCAL as the prologue directs. The user ID you specify in the JCL must be the TSO logon ID dedicated to the Problem Escalation Facility. The user ID must be in the MASTER privilege class because that class has the authority to update all problem records, regardless of which class is the owning privilege class.

Be sure the JOB card matches the defined standards for your installation. You may need a JOBCAT card if the data sets are not in the master catalog. Also, you may not require a STEPLIB if the Information/Management load modules are in the system link pack area.

Information/Management creates a data set called &ZPREFIX, where &ZPREFIX is the ISPF variable that is set when you enter PROFILE PREFIX(user ID). If Information/Management cannot find a value for &ZPREFIX, a data set called...
&ZUSER.$ESTEMP$ is created instead. You can set the variable by using the TSO profile prefix command.

Tailoring the Escalation Facility

An s-word index of 0121 in a RULES record indicates that the prefix collected with it is used as escalation criteria. Escalation processing looks at the p-word associated with each 0121 s-word index and scans the rest of the RULES record for other occurrences of that prefix. If the prefix has data, that data is used in the search argument for escalation processing.

The following example illustrates how an escalation search argument is built.

**Note:** The line numbers in this example are for reference only.

This panel shows an internal view of a RULES record. The s-word index of 0121 for lines 1 through 4 shows the prefixes that are used as escalation criteria.

In this example, the TSP first looks in the current record for an occurrence of the prefix PRI/. It finds PRI/02 (line 9) and adds it to the search argument. The TSP then adds TYPE/PROBTYPE (line 11) because of line 2, and it adds LOCC/LOCATION (line 13) because of line 3 to the search argument. Because of line 4, it looks for COMK/ next. No COMK/ p-word has an associated value, so this escalation criteria is not a part of the search argument in this RULES record.
Tailoring the Escalation Facility

Changing the Escalation Criteria
The escalation facility as shipped from IBM uses the following fields in the RULES records:

- Initial priority
- Problem type
- Location code
- Key item affected

To change the escalation criteria:

- Add the p-word (for example, PERH/) to the dictionary, if it is not already there.
- Update control panel BLG1A410 to include the ADD control lines with the s-word index 0121 and the required p-word index for the criteria you want to include in your search argument.
- Update the Escalation Rules panel (BLG0Z100) to reflect the same criteria. For example, if you add location code as an escalation criterion, add the Location Code field to panel BLG0Z100.

You can also delete criteria in the same way. Although this task is optional if you delete criteria, we recommend that you make the modifications to prevent any confusion about which escalation criteria are used.

- Modify the Summary, Display, Inquiry, Utility, and Delete panels to reflect your changes.
- Modify the RFTs that print RULES records.

Changing the Escalation Command Template
The command used to send an escalation alert message is XMIT. If you are using the BLGESCX6 TSX, the alert message is sent through TCP/IP SMTP mail. If your installation uses a different command or uses a different format, change the escalation command template that is defined by a MOVEVAR control line in TSP BLGESC04 and saved in the ESCB.

The format of the command template is:

```
XMIT (&NODE/&ID) MSGDDNAME(&DDN) NOLOG
```

It can contain one or more of the following variables:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;NODE</td>
<td>Node of the message recipient</td>
</tr>
<tr>
<td>&amp;ID</td>
<td>User ID of the message recipient</td>
</tr>
<tr>
<td>&amp;DDN</td>
<td>ddname of a data set to be sent</td>
</tr>
<tr>
<td>&amp;DSN</td>
<td>Data set name of a data set to be sent</td>
</tr>
<tr>
<td>&amp;MSG</td>
<td>Text of the message</td>
</tr>
</tbody>
</table>

These variables are the only ones for which you can make substitutions. You cannot specify ISPF or system variables in the command template.

Three sample commands follow:
Tailoring the Escalation Facility

SEND '&MSG.' USER(&ID.) LOGON
XMIT &NODE./&ID. MSGDDNAME('&DDN. ') NOLOG
XMIT &NODE./&ID. DA('&DSN. ') NOLOG

Note: Using &DSN with XMIT can degrade performance of the escalation facility. Because XMIT frees the data set automatically when it is finished, the escalation code must re-allocate the data set every time a message is sent.

If you want to use TCP/IP SMTP mail rather than XMIT to send messages, change the command template to:

&ID MSGDDNAME(&DDN)

&ID. The mail alias from the RULES record
&DDN. The ddname to which the escalation data set is written

When the Escalation Facility determines that it is time to send a message, the command template is extracted from the ESCB. All variables are filled in from the RULES record, the TSCA variable data area, and, if specified, other parameters on the call to user exit BLGESNOT in TSP BLGESC06.

Follow all variables with a period (.). If a variable in the template contains a period, two periods must be present (..). For example, if you use USERID.NODE, your command template must contain &ID.&NODE.

Variables that are not specified are not extracted from the RULES record to be put into the command template. For example, using the previous SEND command template sends the message to a user at the node on which the escalation job is running, ignoring the node specified in the RULES record.

Specify only the variables that your SEND command will use.

Changing the Escalation Message

The escalation notification message shipped with the Escalation Facility reads as follows:

ALERT: REC=record id, PRI=current priority, STAT=current status, DESC=problem description.

The BLGESC07 TSP builds this message. The BLGESNOT user exit assembles the string of data to be sent to the ISPEXEC module and substitutes the message into the position of the &DSN. or &MSG. variable in the escalation command template. You can modify the escalation message to meet your requirements either by changing TSP BLGESC07 or by building a message panel including the message you want to send.

You can specify a message panel and extract the message from it to create a simpler one, but you are limited to inserting only one piece of variable data into the message. If your message has inserted data, it is processed in the same manner as in the TSP MESSAGE function, and it requires the same input. Refer to the Terminal Simulator Guide and Reference for more information about the MESSAGE function. Decide which variable you want to insert, such as the RNID, and do the following:
1. Create a message panel containing the new message.
2. Change TSP BLGESC06 by removing its LINK to TSP BLGESC07.
3. Update the USEREXIT line in BLGESC07 that calls BLGESNOT to include the message panel name in the Panel name field.

Using the BLGESCX6 TSX with TCP/IP SMTP

Information/Management provides a TSX that can send escalation notification messages through TCP/IP SMTP rather than through XMIT, using the same logic as the immediate notification TSXs. The BLGESCX6 TSX can be used if you choose to create the escalation message using the existing ddname option. You can modify your escalation processing to build the message as desired and send it through any mail processing available on MVS.

If you use the standard XMIT method to send messages, user IDs and nodes are stored in RULES records that define the search criteria for problems to be escalated, and the people to alert at each escalation level. If you use TCP/IP SMTP, you specify aliases in the user ID fields and nothing in the node fields. The aliases represent the actual mail addresses which are in the USERS record. To use the escalation TSX, you must define a new message skeleton that includes a mail alias. The mail alias is stored in the Level 1, 2, and 3 ID fields of the RULES record. TSX BLGESCX6 extracts mail addresses that are defined for the mail alias from the USERS record. You must define the mail aliases and addresses in the USERS record, or customize escalation processing to determine the mail address using an alternative method.

To use the BLGESCX6 TSX, you must first perform the following steps:

- **Store a mail alias** in the Level 1, 2, and 3 ID fields of the RULES record, instead of the userids and nodes. Then, define the mail aliases in the USERS record in the ASSIGNEE NAME(ALIAS) field on BLGLUSER, the Mail Information Entry panel, and include their corresponding mail addresses.

- **Modify TSP BLGESC04** to define the message skeleton.

\[
&ID:. \text{MSGDDNAME}(&DDN) \\
\]

where \&ID; is the mail alias obtained from the RULES record, and \&DDN; is the ddname of a data set containing the escalation message.

- **Modify TSP BLGESC06**. This TSP calls other TSPs and user exits to build the escalation notification message. It also puts the message in a temporary data set, and issues the ISPF command built by the BLGESNOT user exit. To modify BLGESC06:

  - On the calls to user exit BLGESNOT, specify NOCMD in the USER DATA field on panel BLM8CU9P, Data Field Specification. Specifying NOCMD directs the user exit to substitute for the variables in the message skeleton but not build an ISPF command.

  - After the calls to BLGESNOT, replace the ISPEXEC control lines with LINKS to TSX BLGESCX6.
Modify TSX BLGESCX6. TSXs supplied with Information/Management reside in the SBLMTSX data set. You should copy the TSX to another data set before making modifications. TSXs that Information/Management invokes must be allocated to the BLGTSX DD.

- Define the correct TCP/IP SMTP mail header and trailer information for your environment. See “Defining TCP/IP SMTP Header and Trailer Information” on page 7-7 for more information.
- Define the logical record length, SYSOUT class, destination, and ddname of the data set dynamically allocated to TCP/IP SMTP to match your local environment. See “Defining the SMTP Data Set” on page 7-8 for more information.
- Modify the 'From:' mail address. See “Modifying the "From" Mail Line” on page 7-8 for more information.

**Notification Management User Exits**

The user exits described in this section are intended for use by the escalation facility. Eight of these user exits, BLGESFCB, BLGESGCB, BLGESGET, BLGESPUT, BLGESPUPV, BLGNSYAL, BLGNSYFR, and BLGUSERS are also used by immediate notification. After you understand these exits and how they work together, you can use them for your own purposes.

Two possible input types are fields in the TSCA that are expected to be set before the user exit can run and data that is entered when the USEREXIT control line is created. In the first case, the input is described by using the TSCA field name. In the second case, the field name on the USEREXIT panel is specified. The exception to this is the variable data area, which does not have a TSCA name.

**Note:** User exits which require that information be passed in any field, other than the variable data area (VDA), cannot be run from a TSX. Refer to the Environment section of each of the user exits in this chapter to determine whether or not the user exit can be run from a TSX.

**The Escalation Control Block**

The escalation control block (ESCB) holds anchor information about a rules record while a batch escalation job is running.

Table 14 on page 7-66 describes the contents of the control block; it also includes the field lengths and the offset of each field into the record. This information is provided for your use in diagnosing problems in the escalation facility. These fields must not be changed.
### Table 14. Escalation Control Block Fields

<table>
<thead>
<tr>
<th>Field Number</th>
<th>Description</th>
<th>Field Length (bytes)</th>
<th>Hex Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Escalation control block</td>
<td>2BE</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>ESCB acronym</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Rules record RNID</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>Reserved</td>
<td>2</td>
<td>C</td>
</tr>
<tr>
<td>4</td>
<td>Priority adjust amount</td>
<td>2</td>
<td>E</td>
</tr>
<tr>
<td>5</td>
<td>Adjust direction (+ or -)</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>Reserved</td>
<td>7</td>
<td>11</td>
</tr>
<tr>
<td>7</td>
<td>Level 1 duration</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>8</td>
<td>Level 1 ID to notify</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>9</td>
<td>Level 1 node</td>
<td>8</td>
<td>28</td>
</tr>
<tr>
<td>10</td>
<td>Level 2 duration</td>
<td>8</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td>Level 2 ID to notify</td>
<td>8</td>
<td>38</td>
</tr>
<tr>
<td>12</td>
<td>Level 2 node</td>
<td>8</td>
<td>40</td>
</tr>
<tr>
<td>13</td>
<td>Level 3 duration</td>
<td>8</td>
<td>48</td>
</tr>
<tr>
<td>14</td>
<td>Level 3 ID to notify</td>
<td>8</td>
<td>50</td>
</tr>
<tr>
<td>15</td>
<td>Level 3 node</td>
<td>8</td>
<td>58</td>
</tr>
<tr>
<td>16</td>
<td>Form of command skeleton</td>
<td>50</td>
<td>60</td>
</tr>
<tr>
<td>17</td>
<td>The current date</td>
<td>8</td>
<td>B0</td>
</tr>
<tr>
<td>18</td>
<td>The current time</td>
<td>5</td>
<td>B8</td>
</tr>
<tr>
<td>19</td>
<td>Reserved</td>
<td>3</td>
<td>BD</td>
</tr>
<tr>
<td>20</td>
<td>Length of prefix</td>
<td>1</td>
<td>C0</td>
</tr>
<tr>
<td>21</td>
<td>Actual prefix</td>
<td>10</td>
<td>C1</td>
</tr>
</tbody>
</table>

**BLGESADD–Increment Counter**

Adds 1 to the value in TSCAUFLD. This counter has leading zeros in its format.
Notification Management User Exits

Input

TSCAUFLD as set by a SETFIELD control line
TSCAIPTR containing a pointer to a valid ESCB.

Output

TSCAUFLD is incremented by 1. Possible return and reason codes are listed in Table 15.

Environment

TSP only.

Table 15. BLGESADD Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>A nonnumeric value was specified in TSCAUFLD.</td>
</tr>
</tbody>
</table>

The BLGES01 TSP contains an example of this user exit.

BLGESCCL–Escalation Cleanup

Deletes the escalation load modules and frees the temporary data set.

Input

TSCAIPTR containing a pointer to a valid ESCB.

Output

Escalation load modules are deleted, and the temporary data set is freed. Possible return and reason codes are listed in Table 16.

Environment

TSP only.

Table 16. BLGESCCL Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCB pointer is not valid.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Internal error was detected.</td>
</tr>
<tr>
<td>8</td>
<td>C</td>
<td>Error occurred freeing data set. Data set was not freed.</td>
</tr>
</tbody>
</table>

The BLGESCAL TSP contains an example of this user exit.

BLGESCKE–Check Escalation

Checks to see if an escalation job is in process.

Input

None.
Output  Return and reason codes, as listed in Table 17 on page 7-68.

Environment  TSP only.

<table>
<thead>
<tr>
<th>Table 17. BLGESCKE Return and Reason Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Code (TSCAFRET)</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

The BLGNOTPU TSP contains an example of using this user exit.

**BLGESCLR—Clear Control Block**

Resets the ESCB fields to binary zeroes.

**Input**  TSCAIPTR containing a pointer to a valid ESCB.

**Output**  ESCB is cleared. Possible return and reason codes are listed in Table 18.

**Environment**  TSP only.

<table>
<thead>
<tr>
<th>Table 18. BLGESCLR Return and Reason Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Code (TSCAFRET)</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

The BLGESC03 and BLGESC04 TSPs contain examples of using this user exit.

**BLGESDAT—Date and Time**

Puts the system date and time into the TSCA variable area.

**Input**  The TSCA containing the contents of the **Get variable data** field in the Flag Field Specification panel of the USEREXIT control line.

- **YES**  The date and time is appended to the existing variable data.
- **NO**  The date and time overlay the existing variable data.
Output

TSCAVDAL containing the length of date and time data in the variable data area; variable data area containing the date in the YY/MM/DD format and the time in the HH:MM format. Return and reason codes are listed in Table 19 on page 7-69.

Environment

TSP only.

<table>
<thead>
<tr>
<th>Table 19. BLGESDAT Return and Reason Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Code (TSCAFRET)</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

The BLGESC06 TSP contains an example of this user exit.

BLGESDUR—Duration

Locates the date and time fields and determines whether the duration (1, 2, or 3 depending on TSCAUFLD) specified by the current rules record is met for notifying a user ID. A problem record must be in update mode, and the escalation control block must contain the appropriate duration.

Input

TSCAIPTR containing a pointer to a valid escalation control block.

TSCAUFLD containing the level of current escalation.

The **Structured word index** field in the Data Field Specification panel of the USEREXIT control line to use for comparison for escalation levels 1 to 3 (the **Date entered** field, as shipped for TSP BLGESC06).

The **New Structured word index** field in the Data Field Specification panel of the USEREXIT control line to use for comparing escalation levels (the **Time entered** field, as shipped for TSP BLGESC06).

Output

Return and reason codes as listed in Table 20.

Environment

TSP only.

<table>
<thead>
<tr>
<th>Table 20 (Page 1 of 2). BLGESDUR Return and Reason Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return Code (TSCAFRET)</td>
</tr>
<tr>
<td>------------------------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>
The BLGESC06 TSP contains an example of this user exit.

**BLGESFCB–Free Control Block**

Frees the escalation control block. This must be the last user exit processed in the escalation function.

**Input**

TSCAIPTR containing a pointer to a valid escalation control block.

**Output**

TSCAIPTR is set to 0. Possible return and reason codes are listed in Table 21.

**Environment**

TSP only.

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>Date or time was not specified in the problem record.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCB pointer is not valid.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Internal error was detected.</td>
</tr>
<tr>
<td>8</td>
<td>C</td>
<td>Date or time conversion error.</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>Level number in TSCAUFLD is not valid.</td>
</tr>
</tbody>
</table>

The following TSPs contain examples of using this user exit:

- BLGESCAL
- BLGNOTAE
- BLGNOTAU
- BLGNOTCE
- BLGNOTCU

**BLGESGCB–Get Control Block Storage**

Gets storage for the ESCB and anchors it in the TSCA. This user exit must be called before any other escalation user exit is called.
Notification Management User Exits

**Input** None.

**Output** TSCAIPTR containing a pointer to a valid ESCB. Possible return and reason codes are listed in Table 22.

**Environment** TSP only.

---

Table 22. BLGESGCB Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Escalation control block was set up successfully.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Internal error was detected.</td>
</tr>
</tbody>
</table>

The BLGESCAL TSP contains an example of this user exit.

**BLGESGET—Get Control Block Field**

Gets a field stored by BLGESPUT, BLGESPUV, or a user-written user exit from the ESCB and stores it in the TSCA variable data area.

**Input** TSCAIPTR containing a pointer to a valid ESCB.

The **Literal/Test data** field in the Data Field Specification panel of the USEREXIT control line calling this exit and containing the hex length of the data to be moved.

The **New data** field in the Data Field Specification panel of the USEREXIT control line calling this exit and containing the 1-to 7-byte hex offset within the escalation control block of the data to be moved.

The **Get variable data** field.

**YES** To append the data to the end of the variable data area.

**NO** To replace the data in the variable data area.

**Output** TSCAVDAL containing the length of the data in the variable data area. The variable data area is updated with data from the ESCB. Possible return and reason codes are listed in Table 23 on page 7-72.
Environment  TSP only.

Table 23. BLGESGET Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>The length specified in the Literal/Test data field is not valid.</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>The offset specified in the New data field is not valid.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCB pointer is not valid.</td>
</tr>
</tbody>
</table>

The following TSPs contain examples of this user exit:

BLGESC01
BLGNOTAWE
BLGNOTAU
BLGNOTCE
BLGNOTCU
BLGESC02

BLGESINI–Initialize

Loads the escalation user exits and allocates a temporary data set that you can use with the BLGESNOT user exit to send the escalation message.

Input  TSCAIPTER containing a pointer to a valid ESCB.

Output  Environment for the escalation function.

Environment  TSP only.

Possible return and reason codes are listed in Table 24.

Table 24. BLGESINI Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCB pointer is not valid.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Internal error was detected.</td>
</tr>
<tr>
<td>8</td>
<td>C</td>
<td>An error occurred while allocating data set.</td>
</tr>
</tbody>
</table>
The BLGESCAL TSP contains an example of this user exit.

**BLGESLVL—Level Increment**

Increases the escalation level by 1 in the current problem record ESCB, if possible.

**Input**

TSCAIPTR containing a pointer to a valid ESCB.

**Output**

Escalation level incremented by 1 up to a maximum of 4. No processing occurs if the record has no escalation level or an escalation level of 4. Possible return and reason codes are listed in Table 25.

**Environment**

TSP only.

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>No escalation level field was found in the problem record.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Internal error was detected.</td>
</tr>
</tbody>
</table>

The BLGESC06 TSP contains an example of this user exit.

**BLGESNOT—Notify**

Builds the command that notifies a user about a problem record. The command is built and stored in the TSCA variable data area, overlaying what was there. The message comes from one of the following:

- The message line in an input message panel
- Text in the input variable data area

If NOCMD is specified in the USER DATA field on BLM8CU9P, the Data Field Specification panel, the BLGESNOT user exit does not build an ISPF message; rather, it performs a substitution for the variables in the message skeleton.

To send a message from a message panel:

**Input**

TSCAIPTR containing a pointer to a valid ESCB.

TSCAUFLD containing the level of current escalation set by a SETFIELD control line or the BLGESADD user exit.
The **Panel name** field in the Data Field Specification panel of the USEREXIT control line calling this user exit and containing the message panel name from which to extract a message.

TSCAVDAL containing the length of the send command.

**Output**

TSCAVDAL containing the length of the send command.

Variable data area containing the send command in a format that can be passed directly to ISPEXEC.

**Environment**

TSP only.

To send a message from a string of data:

**Input**

TSCAIPTTR containing a pointer to a valid ESCB.

TSCAUFLD containing the level of current escalation set by a SETFIELD control line in a TSP or the BLGESADD user exit.

TSCAVDAL containing the length of the message to be sent.

Variable data area containing the notification message if no message panel is used.

**Note:** The message must be created by any combination of a MOVEVAR control line, the BLGESGET user exit, and a user-written user-exit routine.

**Output**

TSCAVDAL containing the length of the send command.

The variable data area containing the send command in a format that can be passed directly to ISPEXEC.

**Environment**

TSP only.

Possible return and reason codes are listed in Table 26 on page 7-75.
Notification Management User Exits

The BLGESC06 TSP contains an example of this user exit.

**BLGESPRI—Priority Update**

Increments the value associated with an input prefix by the priority adjust amount in the escalation control block. The record must be in update mode.

**Input**

TSCAIPTR containing a pointer to a valid ESCB.

The **Literal/Test data** field in the Data Field Specification panel of the USEREXIT control line to be escalated (the **Current Priority** field as shipped for TSP BLGESC06).

**Output**

The specified field in the current record is incremented by the specified amount.

**Environment**

TSP only.
Possible return and reason codes are listed in Table 27 on page 7-76.

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Priority was adjusted successfully.</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>Priority adjustment amount equals 0.</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>Priority reached maximum (01 if the priority is being decremented, 99 if the priority is being incremented); priority is not adjusted.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Current priority field (the prefix specified in the <strong>Literal/Test data</strong> field) is not in the problem record; priority is not adjusted.</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>Rules record contains a priority adjustment amount that is not valid; priority is not adjusted.</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>The prefix in the <strong>Literal/Test data</strong> field is not valid, or the data in the problem record for that prefix is greater than two characters.</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>The data in the problem record associated with the prefix in the <strong>Literal/Test data</strong> field is nonnumeric.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCB pointer is not valid.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Internal error was detected.</td>
</tr>
</tbody>
</table>

The BLGESC06 TSP contains an example of this user exit.

**BLGESPUT—Put TSCA Data in Control Block**

Gets data from the TSCA and stores it in the escalation control block. This data can be retrieved by user-exit BLGESGET or a user-written exit routine.

**Input**

- **TSCAIPT**R containing a pointer to a valid ESCB.
- **TSCASDF** containing data (collected by a FINDSDATA control line) to be moved into the ESCB.
The **New data** field in the Data Field Specification panel of the USEREXIT control line containing the 1- to 7-byte hex offset within the ESCB where the data is to be moved.

**Output**
ESCIB updated with the data from the TSCA at the specified offset. Possible return and reason codes are listed in Table 28.

**Environment**
TSP only.

### Table 28. BLGESPUT Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful move.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>The length specified in TSCASDFL is not valid.</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>The offset specified in the <strong>New data</strong> field is not valid.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCIB pointer is not valid.</td>
</tr>
</tbody>
</table>

The following TSPs contain examples of this user exit:

- BLGESCO3
- BLGNOTAE
- BLGNOTAU
- BLGNOTCE
- BLGNOTCU

**BLGESPUV—Put Variable Data**

Gets data from the variable data area or literal data and stores it in the ESCB.

To move variable data:

**Input**

TSCLAIPTR containing a pointer to a valid ESCB.

**Get variable data** field=**YES** in the Flag Field Specification panel of the USEREXIT control line.

TSCLAVDAL containing the length of data to be moved; the TSCA variable data area containing the data to be moved.

The **New data** field in the Data Field Specification panel of the USEREXIT control
line containing the 1- to 7-byte hex offset where the data is moved into the ESCB.

**Note:** Get variable data=YES and **Literal/Test data** are mutually exclusive. If both are specified, data is moved from the variable data area, and **Literal/Test data** is ignored.

**Output** ESCB is updated with the contents of the variable data area.

**Environment** TSP only.

To move literal data:

**Input** TSCAIPTTR containing a pointer to a valid ESCB.

**Literal/Test data** field in the Data Field Specification panel of the USEREXIT control line calling this exit. This field contains the data to be moved into the ESCB.

**New data** field in the Data Field Specification panel of the USEREXIT control line containing the 1- to 7-byte hex offset where the data is moved into the ESCB.

**Note:** Get variable data=YES and **Literal/Test data** are mutually exclusive. If both are specified, data is moved from the variable data area, and **Literal/Test data** is ignored.

**Output** ESCB is updated with the input value.

Possible return and reason codes are listed in Table 29.

**Environment** TSP only.

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>The length specified in TSCAVDAL or in the <strong>Literal/Test data</strong> field is not valid.</td>
</tr>
</tbody>
</table>
The BLGESC04 and BLGESC06 TSPs contain examples of this user exit.

**BLGESSCT–Store Criteria**

Scans the current rules record for an occurrence of escalation criteria (s-word index 0121) and stores its associated prefix in the next available spot in the ESCB.

**Input**

- TSCAIPTTR containing a pointer to a valid ESCB.
- TSCAULFD containing a 1- to 7-byte index of the next available slot.
- TSCARPD containing the prefix of the escalation criteria (collected from running a FINDSDATA control line) to be moved.

**Output**

The next available slot in the ESCB is filled. Possible return and reason codes are listed in Table 30.

**Environment**

TSP only.

---

### Table 29 (Page 2 of 2). BLGESPUV Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>Offset specified in the <strong>New data</strong> field is not valid.</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
<td>Neither variable nor literal data was specified for move (<strong>Get variable data</strong> = NO or blank, and the <strong>Literal/Test data</strong> field is blank).</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCB pointer is not valid.</td>
</tr>
</tbody>
</table>

---

### Table 30 (Page 1 of 2). BLGESSCT Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>The p-word contained in TSCARPD is not valid.</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>The p-word length contained in TSCARPD is not valid.</td>
</tr>
</tbody>
</table>
The BLGESC01 TSP contains an example of this user exit.

**BLGESSEA--Get Escalation Criteria**

Scans the current rules record for an occurrence of data for each escalation criterion, and adds any criteria found to the variable data area as part of a search argument. The search argument locates problem records that meet the escalation criteria in the rules record.

**Input**

TSCAIPTR containing a pointer to a valid ESCB. The control block is valid if the return code is 0.

TSCAUFLD containing the number of criteria in the ESCB.

The TSCA variable data area containing 6,1,5E+ (the base argument for finding problem records).

**Output**

TSCAVDAL containing the length of data in the variable data area.

The TSCA variable data area containing the complete search argument for problem records that match the criteria in this rules record.

Possible return and reason codes are listed in Table 31.

**Environment**

TSP only.

### Table 30 (Page 2 of 2). BLGESSCT Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>C</td>
<td>Limit of criteria fields (30) has been reached.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCB pointer is not valid.</td>
</tr>
</tbody>
</table>

### Table 31 (Page 1 of 2). BLGESSEA Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>TSCAUFLD did not contain the number of criteria in the ESCB.</td>
</tr>
</tbody>
</table>
The BLGESCO1 TSP contains an example of this user exit.

**BLGNSYAL–Allocate Data Set to SYSOUT**

Dynamically allocates a data set to SYSOUT with a user-specified destination. Only one data set can be allocated at a time and must be freed using user exit BLGNSYFR.

**Input**

```
Call BLGTSX
'USEREXIT','BLGNSYAL',ddname,lrecl,class,node,userid
```

- **ddname** is the optional ddname to which to allocate the data set. The ddname is optional. If it is not specified, the system assigns the ddname and it is returned in the REXX variable BLG.NotifyDD.

- **lrecl** is the logical record length to use to allocate the data set (1 to 3 digits). This is a required parameter.

- **class** is the SYSOUT class to which to allocate the data set (1 character). This is a required parameter.

- **node** and **userid** represent the SYSOUT destination. For immediate notification and escalation, this targets the TCP/IP SMTP server on the system where the server resides. The node is required and it must consist of 1 to 8 characters. The userid is optional and, if specified, must consist of 1 to 8 characters. If the userid is not specified, it defaults to SMTP.

**Output**

A SYSOUT data set is allocated using the specified inputs and a variable (V) record format. If a SYSOUT data set was previously allocated and not freed using BLGNSYFR, nothing occurs.

---

**Table 31 (Page 2 of 2). BLGESSEA Return and Reason Codes**

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>8</td>
<td>Variable data area is full.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>ESCB pointer is not valid.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Internal error was detected.</td>
</tr>
</tbody>
</table>

---

7-81
Environment  TSX only.

Possible return and reason codes are listed in Table 32.

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Processing successful, SYSOUT data set allocated.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>SYSOUT data set is already allocated and not freed with BLGNSYFR. The data set remains allocated.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>BLGNSYAL was called from a TSP. Only a TSX can call it.</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>Logic error. Internal control blocks could not be located. No addresses are returned.</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>No parameters were specified.</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>The DD name length is not valid.</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>The LRECL was either not specified, greater than 3 characters, or not numeric.</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>The SYSOUT class was not specified or it is more than 1 character.</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>The node was not specified or the length is not valid.</td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>The user ID length is not valid.</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>An error occurred while allocating the SYSOUT data set. Additional messages may have been issued that identify the specific problem.</td>
</tr>
</tbody>
</table>

The BLGTXNOT TSX contains an example of this user exit.

**BLGNSYFR—Free Data Set Allocated by BLGNSYAL**

Frees the SYSOUT data set allocated by user exit BLGNSYAL.

**Input**  
Call BLGTSX 'USEREXIT', 'BLGNSYFR'

**Output**  
The SYSOUT data set is freed.

**Environment**  
TSX only.

Possible return and reason codes are listed in Table 33 on page 7-83.
The BLGTXNOT TSX contains an example of this user exit.

BLGUSERS—Extract Mail Address from USERS Record

Extracts the mail addresses from the USERS record that match an input name or mail alias.

**Input**

Call BLGTSX

'USEREXIT','BLGUSERS',name,databaseid,'SAVE' or 'NOSAVE'

- **name** - The assignee name or mail alias that is defined in the USERS record. The name is required and must consist of 1 to 60 characters. If more than 60 characters are specified, the name is truncated to 60 characters.

- **databaseid** - An optional single number identifying the database from which to read the USERS record. If more than 1 character is specified only the first character is used. The default is the number 5.

- **SAVE or NOSAVE** - Saves the USERS record in storage, or does not save the record in storage. This parameter is required.

---

### Table 33. BLGNSYFR Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Processing successful, SYSOUT data set is freed.</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>SYSOUT data set was not previously allocated by user exit BLGNSYAL.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>BLGNSYFR was called from a TSP. Only a TSX can call it.</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>Logic error. Internal control blocks could not be located. No addresses are returned.</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>An error occurred while freeing the SYSOUT data set. Additional messages may have been issued that identify the specific problem.</td>
</tr>
</tbody>
</table>

---

The BLGTXNOT TSX contains an example of this user exit.
Output

Addresses are written to REXX stem BLG_ADDRESS. BLG_ADDRESS.0 contains the number of addresses returned. The maximum number of addresses that can be returned is 200.

Environment

TSX only.

Possible return and reason codes are listed in Table 34.

Processing Notes:

You can specify that the USERS record be maintained in storage and not re-read on each call, unless another user updates it. If the user exit is called with the NOSAVE option the record is not maintained in storage and is re-read each time BLGUSERS is invoked. The NOSAVE option directs BLGUSERS to unconditionally read the USERS record and not maintain it in storage. By using BLGUSERS, your TSX does not have to explicitly display the USERS record in order to extract mail addresses from the list of names and addresses.

BLGUSERS:

- Assumes the USERS record name/alias and address lists are sorted by name or alias. All entries with the same name must be together in the list.
- The name or alias length must be less than or equal to 60 characters.
- The address list in the USERS record must be defined as either string data or a maximum response of one word.
- Each name or alias in the USERS record must have a corresponding address, and vice versa.

If desired, TSXs can also access the USERS record data through normal TSP processing (by displaying the record and using FINDSDATA and/or GETLIST to search for and retrieve data).

Table 34 (Page 1 of 2). BLGUSERS Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Processing successful, at least one address is returned.</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>At least one name in the list did not have a corresponding address. At least one address is returned.</td>
</tr>
</tbody>
</table>
### Table 34 (Page 2 of 2). BLGUSERS Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8</td>
<td>The number of addresses found exceeds the maximum allowed. The maximum number of addresses is returned.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>The USERS record was not found. No addresses are returned.</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>The input name was not located in the USERS record or no address was defined for the first name occurrence. No addresses are returned.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>Out of storage. No addresses are returned.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>BLGUSERS was called from a TSP. Only a TSX can call it.</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>Logic error. The internal control blocks could not be located. No addresses are returned.</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>No parameters were specified or the assignee name or mail alias was not specified. Name and SAVE/NOSAVE are required. No addresses are returned.</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>The action parameter must be SAVE or NOSAVE. No addresses are returned.</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>First character of the name parameter was a blank. Remove leading blanks. No addresses are returned.</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>An error occurred while reading the USERS record. Other messages may be issued that indicate the specific error that occurred. No addresses are returned.</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>The specified or default database could not be accessed. No addresses are returned.</td>
</tr>
</tbody>
</table>

The BLGTXNOT TSX contains an example of this user exit.
Defining User Line Commands

This chapter contains Product-sensitive Programming Interface and Associated Guidance Information.

Using PMF, you can define line commands that are unique to your installation and that can be used on various search results lists. You can also define commands that process when a user issues the RUN command. Both cases of user-defined commands are dependent upon the running of TSPs or TSXs. You can make your user’s job easier by creating alternate names to identify these TSPs and TSXs. For example, you could create a TSP called TSPCLSRD or a TSX called TSXCLSRD to close a record. Most users probably find it easier to remember something simple, like “close,” than to memorize 8-character TSP (or TSX) names. You can relate the simple name “close” to your TSP or TSX by using the Information/Management ALIAS record.

This chapter tells you how to define a user line command for use on a search results list, create an ALIAS record, add an entry to the ALIAS record, and write a TSP or TSX to process a user line command. You must understand Terminal Simulator Panels (TSPs), Terminal Simulator EXECs (TSXs), and the Panel Modification Facility (PMF) to perform the TSP- and TSX-related tasks described in this chapter. For information about TSPs and TSXs, refer to the Terminal Simulator Guide and Reference. For information about PMF, refer to the Panel Modification Facility Guide.

User line commands can also be defined for use on list processor panels. Refer to the Panel Modification Facility Guide for all information regarding list processor user line commands.

Defining User Line Commands for Search Results Lists

You can define unique line commands to use on search results lists. For example, you can define a line command to automatically close a problem record. Currently, a user must select the record for update from a search results list, change the Status field to CLOSED, and then file the record. You can automate this procedure by defining a user line command.

When a user enters a valid user line command on a search results list, Information/Management takes the data on the table display and puts it into a linked list with any other data collected by previous user line commands. User line commands are not immediately processed. The delay in running the commands
Defining User Line Commands for Search Results Lists

allows for multiple records to be used for a single purpose. For example, the user can select a list of features to add to a component record, or the user can close a group of records.

Once the user line command data is collected, either the user or the installation can determine when to process it. Two commands, RUN and DROP, work with the user line commands. Refer to the User’s Guide for complete information about these commands.

**Defining User Line Commands and Where They Are Valid**

The first thing that you must determine is what line commands you want to define, and on which search results list panels you want to use the new line commands. You must also determine the names of any TSPs or TSXs that you want to run when a user enters one or more of these new commands. Once you have this information, define the new commands as follows:

1. Create a p-word entry in the dictionary for each new user line command. Do not specify a prefix, but do specify the data validation as the line command. For example, if you want to define three user line commands (X, Y, and Z), the data validation for each is as follows:

<table>
<thead>
<tr>
<th>Command</th>
<th>Data Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>&lt;X&gt;</td>
</tr>
<tr>
<td>Y</td>
<td>&lt;Y&gt;</td>
</tr>
<tr>
<td>Z</td>
<td>&lt;Z&gt;</td>
</tr>
</tbody>
</table>

For this example, the Dictionary Display panel appears like this.

```
 prefix  prefix  data validation  pgm used  general comment
     index  word     validation     used   command

    8004 <X> NO USER LINE COMMAND
    8005 <Y> NO USER LINE COMMAND
    8006 <Z> NO USER LINE COMMAND

Line Cmds: A=Add  D=Delete  R=Repeat
Type DOWN, UP, LEFT, or RIGHT to scroll the panel, or type END to exit.
```
2. Create a control panel called BLG1AA01. Add one line to this control panel for each line command and search results list combination. If you have two search results list panels that each allow three user line commands, you must create six control lines in the BLG1AA01 control panel. Use a FLOW line with a function code index of 002A, a true target of the search results list panel name, and a p-word index of one of the line commands for each combination.

If you want to specify a TSP or a TSX to process this command when the user leaves the search results list, make the false target of this control line the name of the TSP or the TSX.

**Note:** *Leaves*, in this context, means that the user enters END, CANCEL, QUIT, INITIALIZE, or RESUME. A search results list is still in effect if the user can return to it at a later point in time. If a user does a SUSPEND, the search results list is still active because the user can later issue a RESUME to return to the original list. The user can also use the DROP PANEL command to drop pending line commands from a previous suspension level. This can only be done if the user has not issued user line commands in the current suspension level. If new line commands were entered in the current suspension level, only those user line commands are affected by the first DROP PANEL command. A second DROP PANEL can be issued to drop the line commands that exist on previous suspension levels instead of using the DROP ALL command. DROP ALL always deletes all data currently collected.

When the user leaves a search results list, the BLGTENDS and BLGTENDL TSPs are run. If you do not need any processing when you leave a search results list, you can delete one or both TSPs from your base panel data set. BLGTENDL runs the DROP PANEL command to free the SLB and all SRB control blocks that were collected to run the user line commands for the current search results list table.
Here is an example of the BLG1AA01 control panel.

<table>
<thead>
<tr>
<th>BLG1AA01 FUNCTION LINE SUMMARY</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNC  FUNC  S-WORD  PREFIX  MULT  APPLY  AUTH  TRUE  FALSE</td>
</tr>
<tr>
<td>TYPE  CODE  INDEX  INDEX  B  E  NOT  CODE  TARGET  TARGET</td>
</tr>
</tbody>
</table>

```
FLOW 002A 0000 8004 N N N 0000 USERTBL2 USERTSP4
FLOW 002A 0000 8005 N N N 0000 USERTBL2 USERTSP5
FLOW 002A 0000 8006 N N N 0000 USERTBL2 USERTSP6
FLOW 002A 0000 8004 N N N 0000 USERTBL3 USERTSP60
FLOW 002A 0000 8004 N N N 0000 USERTBL1 TESTTSP1
```

Note the following about the control panel:

- The prefix index listed in the control panel is the same as the prefix index in the dictionary entry for each user line command.

- The **True target** field identifies each search results list panel that the user line command is valid on.

If a user line command is valid on 3 different search results list panels, you need 3 FLOW control lines for each user line command and search results list panel combination. Information/Management uses the information specified in BLG1AA01 to insure that only user line commands associated with a particular search results list panel are valid on that panel. An optional TSP or TSX, listed in the **False target** field, can also be associated with each user line command and search results list panel combination.

In this example, if a user issues user line command Y on a search results list, the second FLOW control line (for prefix index 8005) is referenced. If the user issues the line command from the panel listed in the **True target** field (USERTBL2), TSP USERTSP5 runs when the user leaves the search results list or issues the RUN command. If the user does not issue the line command from the panel listed in the **True target** field, Information/Management scans the rest of the control panel for a match. If there is no match (either the search results list panel was not associated with the user line command or the user line command itself was not listed in the control panel), an error message is issued. If a match is found, but there is no TSP or TSX specified, the BLGTDFLT TSP runs.

3. For additional control lines that cannot fit on this one control panel, you can continue by creating control panel BLG1AA02, then BLG1AA03, and so on up to control panel BLG1AA09. If you skip a panel in the numerical sequence, any panels that follow the omitted number are ignored.
Using TSP and TSX Aliases

When the user line command is processed can be determined by either the end user or the installation, but how the data is processed must be determined by the program administrator.

If your end users prefer to have user line commands process before the search results list ends, they can use the RUN command to start the line commands that have collected so far. Rather than require all your end users to remember a TSP name or a TSX name, you might want to give a more meaningful name to the data to be processed. For example, if you have a user line command to close a record, the user can remember that the command to process that record is RUN CLOSE, rather than RUN TSPCLSRD or RUN TSXCLSRD. This simple name for your TSP or TSX is called an *alias*. You can have more than one alias for each TSP or TSX.

If you want to use an alias, you must add an entry to the ALIAS record. This record correlates the name of your alias to its associated TSP or TSX. Also, you can have an entry in the ALIAS record for a TSP or TSX that has no correlation to user line commands. This type of alias can also be called by the RUN command (for example, RUN CLOSE).

For performance reasons, you should consider creating an alias for each TSX even if the alias name is the same as the TSX name. Refer to the *Terminal Simulator Guide and Reference* for more information on invoking TSXs.

The next section tells you how to create the ALIAS record and add entries to it.
Creating the ALIAS Record

To create the ALIAS record, select **Entry** from the Primary Options Menu for the System application. Type 5 and press Enter.
On BLG00010, the System Record Entry panel, select the kind of record you want to create.

For this exercise, select **Reference**. Type 2 and press Enter.
BLG00020, the Reference Entry panel, offers you a choice of creating three types of reference records. Choose Alias. Type:

2

and press Enter.

If you already have the ALIAS record in your system, this selection brings your existing record into update mode so you can add or change data.

```
+ 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.

+------------------------ SELECT ITEM -----------------------------+

===>
```
If you are creating the ALIAS record for the first time, an empty Alias Name Entry panel appears. On this panel, list the alias names for your TSPs and TSXs. For example, to identify CLOSE as an alias for TSPCLS RD or TSXCLS RD, enter these values into the record. In the **TYPE** column, enter a **P** if this is for a **TSP** or an **X** if this is for a **TSX**. After you enter the data, type `end` and press Enter to save your entries.

```
BLGLALIS ALIAS NAME ENTRY LINE 1 OF 22
USE....List alias name, actual name, and type (panel or REXX EXEC).
RECORD: ALIAS

<table>
<thead>
<tr>
<th>ALIAS NAME</th>
<th>ACTUAL NAME</th>
<th>TYPE</th>
<th>ALIAS NAME</th>
<th>ACTUAL NAME</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>close</td>
<td>tspclsrd</td>
<td>P</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
```
Creating the ALIAS Record

A valid alias name is a string of 1 to 8 SBCS alphanumeric characters. You can enter as many alias names as you need. If you run out of lines on the screen, you can use the R or I line commands to add lines to the record.

BLG0EN10, the Primary Options Menu for the System application, appears with a message that the ALIAS record was stored successfully.

```
BLG0EN10 --- PRIMARY OPTIONS MENU --- APPLICATION: SYSTEM

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. DBADMIN.......Perform database administration.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997

BLG03058I Record ALIAS was stored successfully.

===>
```

Processing User Line Commands

After you complete the setup identified in the previous sections of this chapter, you must write the TSPs or TSXs and user exits to process the user line commands that users enter. If you define many user line commands, but only write TSPs or TSXs that account for a subset of those commands, then you can only use that subset, or your TSP or TSX must be general enough to process all the valid line commands it receives. Each user line command must be processed by a TSP or by a TSX.

To write the TSP or TSX and user exit, you must know the data that the user line commands collect. Three control blocks are involved in the data collection:

- **Search User Block (SUB)**. An SUB is created the first time the user issues a user line command on a search results list. It is used to find your way through the second type of control block, the Search List Buffer.

  The TSCASUBP field in the Terminal Simulator Communications Area (TSCA) points to the SUB control block.
Search List Buffer (SLB). Each time the user issues the SEARCH command, there is one occurrence of the SLB where user line commands were issued. The SLB gives you information about the search results list panel on which the user line commands were issued, and points you to the third type of control block, the Search Row Buffer.

Search Row Buffer (SRB). An SRB is created for each row on which a user line command is entered. Each time a user enters a user line command on a search results list, Information/Management takes all the data that appears on the table display and puts it into an SRB. It then links that SRB to any others that were created by previously entered user line commands on this same table panel. The data in the SRB includes the entire length of the data specified as the maximum for each field in the table panel up to a maximum of 64 characters. This means you get all the data, not just that which appears on the table display.

The SRB also contains the name of the user line command. This is so you can write a user exit that can determine how to use the data based on the line command that the user entered.
Processing User Line Commands

SRB DSECT //Search row buffer
   DS 0F
SRBACRON DS CL4 /'SRB'
SRBFWDP DS A /Pointer to the next SRB
SRBPRVP DS A /Pointer to the previous SRB
SRBLCOL DS A /Pointer to the last column of data
SRBRNID DS CL8 /Record ID of selected row
SRBCPRC DS A /Pointer to last row processed
SRBLCMD DS CL1 /Line command to be processed
   DS CL3 /Reserved field
SRBTSPIN DS F /Index of TSP name in TUB
SRBLNUM DS F /Line number on search results list
SRBIBMIF DS XL1 /Flag field for IBM use
SRBUSRF DS XL1 /Flag field for user data
SRBSHRF DS XL2 /Flag field for shared data
SRBRECFC EQU X'80' /Indicates record has been modified
SRBSDATA DS OCCL68 /Column data for this row
   /*******************************************************************************/
   /* There is one SRBITEM per column of data on a table panel. The */
   /* first column is the line command column, followed by the next */
   /* column of data you defined. For example, on panel BLGITSRL, */
   /* the first SRBITEM is the line command column. The second */
   /* SRBITEM is the Record ID column, and the last SRBITEM is the */
   /* Description Abstract column. */
   /******************************************************************************/
SRBITEM DSECT /*A column of data
   SRBDLEN DS XL2 /Used length of this column
   SRBFLEN DS XL2 /Entire length of this column
   SRBDATA DS CL64 /Actual data for this column

Note: Neither SRBDLEN nor SRBFLEN will ever exceed 64 characters, even if the table panel field length is defined as greater than 64 characters.

The assembler mappings for these control blocks are shipped with Information/Management in the SBLMMACS data set and are named:

   Search User Block        BLGUSUB
   Search List Buffer       BLGUSLB
   Search Row Buffer        BLGUSRUB.

Example User Exit

The sample library (SBLMSAMP) contains an example of an assembler user exit that returns the RNID of a record. This user exit is called BLGSRNID. This example contains a conversion from hexadecimal to printable format. You can use this example in conjunction with the TSCA variable data area.

TSP and TSX Considerations

You have access to the SUB whenever you are running a TSP or TSX because the TSCA points to it. The TSP or TSX you write to handle the user line commands must be coded so they can be called in any of these ways:

- The user can issue the RUN command. If you allow your users to select multiple items from the search results list, you must remember that they might issue the RUN command after each one of them. You do not want your TSP or TSX to remove the user from the search results list in that instance.
- The collected data remains on the system until a DROP command is issued. You can have a TSP or TSX process the data later in your session, or your TSP or TSX itself can issue the DROP command if you no longer need the data.

- When the user leaves the search results list, a TSP or TSX runs. This TSP or TSX is the one identified by the control panel containing the appropriate table panel and user line command combination, if specified.

  **Note:** If you do not assign a TSP or TSX for this particular combination, a default TSP (BLGTDFLT) runs. The BLGTDFLT that IBM ships contains only a RETURN. You must modify BLGTDFLT to work the way you want it to.

- A TSP runs before all other line command TSPs run. This TSP (BLGTENDS) that IBM ships contains only a RETURN. You must modify it if you need special processing after the other line commands are finished. For example, if a previous TSP issued a SUSPEND command, then performed the actual search and issued line commands, you could have TSP BLGTENDS issue a RESUME command.

- A TSP runs after all other line command TSPs are finished. This TSP, BLGTENDL, issues a DROP PANEL command.

  **Note:** IBM-defined line commands take precedence over user-defined line commands. For example, if you create a U line command for a panel where U is already a valid line command, your user-defined value is not recognized.
This example TSP closes a problem record. This TSP can be used with the user exit described in “Example User Exit” on page 8-12.

To further explain the code lines of the TSP in the previous panel:

**Line**  |  **Action**  
--- | ---  
2  |  Starts processing and provides the label to loop back to.  
3  |  Sets the **TSCAUFLD** field to **Y**. The **GETSRNID** user exit, which is run in the next step, uses this value to determine whether to use or skip the SRB created for a particular line command. In this case, it only uses an SRB if it was created because the user specified the **Y** user line command.  
4  |  Starts the **GETSRNID** user exit. (This is the example user exit that is in the sample library and referenced in “Example User Exit” on page 8-12.) It scans through the SRBs for the first one associated with the **Y** user line command that has not yet been processed and stores the corresponding record ID in the TSCA variable data area.  
5  |  This line checks the return code set in the user exit. If the return code is 4, the TSP branches to label **FINIS** and processes a return indicating that all processing is finished.
6 This line issues an UPDATE command. It also performs a GET VARIABLE DATA operation to add the RNID of the record (that the line command was issued for on the search results list) to the command. The RNID is put into the variable data area of the TSCA by the user exit.

7 Processes line 6.

8 and 9 Closes the record and files it.

10 This line branches back to label LOOP and starts the process over again if more line commands need to be processed.
If your end users have a need to enter commands that are meaningful to them (for instance, STARTOVER may be easier to remember than INITIALIZE), you can set up a command alias to customize Information/Management commands and improve user productivity. A command alias is set up by the program administrator through a reference record called the COMMAND record. A command alias can be defined for Information/Management commands or for immediate response chains (IRCs) that begin with an Information/Management command (for example, %PDFEDIT could be set up as an alias for PROFILE,1,51,PDF,,9). Command aliases are invoked the same way as standard Information/Management commands. Anywhere an Information/Management command can be invoked, a command alias can be invoked.

Additionally, you can define an authorization code for each command alias to restrict its usage by privilege class. Command alias and authorization can also be used to set up authorization for the Information/Management commands. To set up authorization for an Information/Management command, define a command alias identical to the Information/Management command and set an authorization code for it.

You can also optionally specify a help panel to be shown when the user enters HELP for the command alias. If you do not specify a help panel, Information/Management displays the help panel for the first command in the actual command string that the command alias represents.

Command alias names can be up to 32 characters long. The actual command string represented by the command alias can be up to 80 characters long. Command alias names must begin with A-Z, $ (X'5B'), # (X'7B'), @ (X'7C'), or % (X'6C'). Other characters in the name can consist of A-Z, 0-9, $, #, or @. The actual command string must begin with a standard Information/Management command.

Future commands or aliases may be added to the Information/Management product which may be identical to a command alias you have defined for your users. To avoid this conflict, start your command alias names with a $, #, @, or % character. Information/Management will not use these characters as the first character of a command.

If you define a command alias to match an Information/Management command exactly, the command alias will be processed in all cases except when it is the first response in the command string for another command alias. The
Information/Management command will always be used as the first response in the actual command string. For example, if a command alias of SHOW is defined with an actual command string of DISPLAY R, the Information/Management DISPLAY command is used even if there is a command alias with a name of DISPLAY defined. This exception enables you to define command aliases which redefine existing Information/Management commands to include additional arguments, or simply to authorize the command aliases. You could, for instance, define a command alias named VIEW for the Information/Management VIEW command, and set it up with a valid authorization code to restrict use of the VIEW command in your user community.

To invoke a command alias, users simply enter the command alias name as they would any other command. Information/Management processes the actual command string for that command alias or displays a message if the user is not authorized.

The next section shows you how to create a COMMAND record, and set up authorization codes for command aliases.

Creating the COMMAND Record

To create the COMMAND record, select Entry from the Primary Options Menu for the System application. Type:

5

and press Enter.
On BLG00010, the System Record Entry panel, select the kind of record you want to create.

For this exercise, select Reference. Type:

2

and press Enter.

---

USE....Identify the type of description (record) to be entered.
1.CLASS............Define authority and users in a privilege class record.
2.REFERENC.........Define reference information.
3.LOGSAVE..........Define information used by the Automatic Log Save and DB2 Extract Facilities.
4.MAP................Define PostProcessor data mapping.

---

---

2
BLG00020, the Reference Entry panel, offers you a choice of creating three types of reference records. Choose Command. Type:

3

and press Enter.

If you already have the COMMAND record in your system, this selection brings your existing record into update mode so you can add or change data.
If you are creating the COMMAND record for the first time, an empty Command Alias/Authorization Entry panel appears. On this panel, list the **Command or Alias Name** for the commands you want to set up with aliases and specify the actual command string the alias represents. In this example, `STARTOVER` is identified as an alias name for the `INITIALIZE` command. **Auth Code** or **Help Panel** entries are optional. After you enter the data, type:

```
end
```

on the command line and press Enter to save your entries.

<table>
<thead>
<tr>
<th>Command or Alias Name</th>
<th>Auth Code</th>
<th>Help Panel</th>
<th>Actual Command String</th>
</tr>
</thead>
<tbody>
<tr>
<td>%STARTOVER</td>
<td></td>
<td></td>
<td><code>INITIALIZE</code></td>
</tr>
<tr>
<td>%SHOW</td>
<td>HLP0BD01</td>
<td></td>
<td><code>DISPLAY R</code></td>
</tr>
<tr>
<td>VIEW</td>
<td>8040</td>
<td></td>
<td><code>VIEW</code></td>
</tr>
</tbody>
</table>

Use **Auth Code** to associate an authorization code with the command alias. The authorization code consists of up to 4 digits. (For more information on how to create authorization codes, refer to the *Panel Modification Facility Guide.*) Use **Help Panel** to associate a help panel with the command alias. The help panel identifier consists of 8 alphanumeric characters.

**Note:** Only part of the **Command or Alias Name** and **Actual Command String** are shown on this panel. The **Command or Alias Name** can be 32 characters long. The **Actual Command String** can be 80 characters long so only part of the fields are actually shown on this panel. You can use the **RIGHT** and **LEFT** commands to scroll the panel. You can enter data which is longer by typing **L1** (for alias name) or **L4** (for actual command string) in the line command area to the left of the row you want to update. Then press Enter and type the field value on the assisted-entry panel. (You cannot, however, enter commas on the assisted-entry panel using this method.)
Creating the COMMAND Record

You can enter as many alias names as you need. If you run out of lines on the screen, you can use the I or R line commands to add lines to the record.

The Primary Options Menu panel appears with a message indicating the COMMAND record was stored successfully.

To change the COMMAND record once it is saved, simply enter 5,2,3 from the System Primary Options menu (BLG0EN10) or use any supported method of record update (e.g., the UPDATE command).

Usage Notes

- Users should enter command aliases the same way they enter normal commands. Any operands the user supplies are appended to the end of the actual command string associated with the alias, separated by a blank. For example, if SHOW is an alias for DISPLAY R, then SHOW 1234 will be processed as DISPLAY R 1234.

- Changes to the COMMAND record become effective the next time the user session interacts with the BLX-SP.

- If you set up a command alias with an actual command string that does not begin with a unique valid Information/Management command, users will receive a message indicating the command is not valid when they invoke the alias name.

- Users and TSPs can specify a command using the minimum number of characters necessary to uniquely identify the command (e.g., SE can be used for SEARCH or Q can be used for QUIT). If a command alias begins with the same characters as an Information/Management command or another command alias, it could change the minimum number of characters needed to
uniquely identify the command, and require that the user or TSP use a longer abbreviation of the command. To avoid having your command alias names conflict with existing commands, refer to the User’s Guide for a list of Information/Management commands and their acceptable truncations.

Future commands or aliases may be added to the Information/Management product which may be identical to a command alias you have defined for your users. This situation could cause the minimum abbreviation requirements to change. **To avoid this possibility, start the command alias names with a $, #, @, or % character.** Information/Management will not use these characters as the first character of a command. Additionally, you should use full command names in a TSP, TSX, or PF key definition to avoid the need to change them should the minimum abbreviation become longer in the future.

- You can use the PRINT command to print the COMMAND record.
Setting Up Stored Response Chains

You can establish a set of stored response chains (SRCs) that enable users to perform repetitive tasks quickly, if you are the program administrator or other authorized user. Also, you can establish SRCs that enter certain data in formats that are appropriate for your organization.

Because an SRC follows the same prompting sequence that a user follows when entering data, you can build an SRC for any prompting sequence that a user follows repeatedly. In addition to being faster, creating SRCs ensures that users follow a particular prompting sequence and enter certain data values. An SRC prompting sequence encourages a consistent method of data entry that helps ensure users can retrieve data as they need it.

Assessing Needs for SRCs

Before setting up SRCs for your organization, assess the needs of each user group. Base your decision on such factors as user experience, need for flexibility and speed, and your guidelines for data collection. If, for instance, each user deals with different types of problems or changes, organization-wide SRCs cannot provide the flexibility your site requires. For greater flexibility and usability, encourage users to design their own SRCs.

It might be useful to establish meaningful naming conventions and good descriptions for SRCs so users can recall the name of the SRC they want to use, or can recognize it in a list of SRCs. For instance, all SRCs for creating records might start with C, followed by the user’s initials, followed by a specific identifier.

Planning SRCs

If you have authority, you can create SRCs for all your Information/Management users. An SRC record consists of two parts: descriptive information and SRC responses. For each SRC that you create, you must decide which responses you want entered by the SRC and which response you want entered by the user.

Before creating an SRC, ask yourself:

- Is there a related SRC that I could use as a guide for creating my new SRC?
- From which panel should the SRC start? Keep in mind that SRCs starting on the Primary Options Menu can also be processed when Information/Management is started, thus bypassing the display of that panel.
• Should the SRC be able to start from any application? You can start an SRC with ;;SUSPEND or ;;INITIALIZ and subsequently select a specific application so that the SRC can be used with any application. Or, if you use only one application, the Invocation application field in the profile can remain set to that application. If you start the SRC with the SUSPEND command, use the RESUME command as either the last command in the SRC (input as ;;RESUME), or in the final response (input as RESUME).

• Should the SRC be able to start from any panel? By beginning an SRC with certain commands (REPORT, for example), any user can, from any panel, use SUSPEND to temporarily leave the current dialog. Then the user can perform searches, print reports, or change the output destination.

■ Should you change the responses of the SRC? You can display and modify only responses within a dialog that begins within the SRC with the CHANGE command. You cannot display or modify any responses before the first collected dialog with the CHANGE command if an SRC begins in the middle of a dialog.

■ Will the SRC run differently based on a user’s profile setting? If so, start the SRC with the PROFILE command and change the settings that affect the processing of the SRC. Temporarily save the profile and end the SRC with PROFILE,8 to restore the user’s profile to the usual settings.

■ Will a search argument in the SRC yield different results based on the length of the SDIDS key? If an SRC contains a fully qualified search argument written for an SDIDS with an 18-byte key, the search results may be different with an SDIDS with a 34-byte key. To get the same results with SDIDSs with both the shorter and longer keys, add a period to the end of the fully qualified search argument written for an SDIDS with a 34-byte key. Information/Management will ignore the characters following the period if any exist in the data record. Refer to the Planning and Installation Guide and Reference for more information about the lengths of SDIDS keys.

■ What data should the SRC provide? Can any common information be entered by default without limiting this SRC to only a few users?

■ What data should the user provide? At what points in the prompting sequence should the SRC stop and let the user enter data that is unique to the particular task, such as the user’s name or an ABEND code?

■ What panel should be displayed to the user when the SRC finishes processing? This panel can be a search results list or a panel that requests additional data.

■ Is this SRC flexible enough for a number of users and yet specific enough to be of value?

■ Which users can use the SRC? The privilege class of the user creating the SRC is automatically entered as the only class that can use the SRC. If you want other users to use the SRC, either clear the privilege class field to authorize
all users, or add each privilege class on the SRC Execute Classes panel. Master privilege class automatically has the authority to use any SRC.

For information on how to create SRCs, refer to the *User’s Guide*.

**Modifying and Controlling SRCs**

To meet more specific or changing needs, you can modify the SRCs created at your organization. To make sure that you can modify the SRCs you create, document each one the same way:

- Use the description of the SRC that you are modifying as a guide to its contents.
- Note such specifics as the name, the starting panel, and the default data.
- Decide which responses in the SRC you want to change.

To modify an SRC, you need SRC update authority. Refer to the *User’s Guide* for information on how to update an SRC. After you make all your changes, file the SRC.

**Rules for Modifying SRCs**

When you modify SRCs, you cannot do the following:

- Add new panels to the prompting sequence
- Change freeform text
- Modify information within a dialog that began before running the SRC

When you modify SRCs, you can do the following:

- Change data in the existing prompting sequence in the SRC and data in the final responses at the end of the SRC.
- Delete commands and the ;;commands collected, not change them. If you delete the ;;PROFILE or ;;REPORT command, the system deletes all responses associated with the command.

If the changed item does not alter the subsequent panel flow, the data you specify replaces the previous data, and the Changed Collected Data panel reappears. If the changed item alters the subsequent panel flow, a warning panel appears asking you to verify the change in the panel flow.

**Deleting SRCs**

Instead of modifying an existing SRC, you can delete it and create a new one. SRCs, like other records, are deleted with the DELETE command or D (delete) line command on a search results list. Refer to the *User’s Guide* for information on working with records and deleting them. Only users with SRC delete authority can delete SRCs.
Controlling SRCs

You can control who creates, updates, and deletes SRCs through privilege class authorities and record ownership. Only users with the appropriate authority can perform these functions. This control is useful in limiting the number of SRCs created and in identifying who is responsible for maintaining them.

Once created, SRCs are available for use by authorized users or by all users if no SRC-execute classes are specified. A user does not need authority to display or print SRCs. A user can display descriptive information about the SRC (such as the SRC description, the start panel name, or the application) and the actual responses in the SRC. The user can also find an appropriate SRC by using information from the description field in a search argument. If a user specifies the EXECUTE command with no operands, he sees a list of SRCs that he is authorized to run from the panel on which he gave the EXECUTE command.
Propagating Information/Management Data

This chapter contains Product-sensitive Programming Interface and Associated Guidance Information.

Information/Management provides the Automatic Log Save Facility, the DB2 Extract Facility, and the LOGSAVE record so that you can propagate your Information/Management production data to other Information/Management or DB2 databases, or to both Information/Management and DB2 databases.

This chapter provides an introduction to data propagation in the Information/Management environment and gives a high-level overview of the Automatic Log Save Facility and the DB2 Extract Facility. It includes a detailed description of the LOGSAVE record and a discussion of synchronization.

Understanding Data Propagation

Data propagation involves applying the changes made to one set of data to the copy of that data in another database system or to another database in the same database system. Thus, multiple sets of data can be kept consistent.

In the Information/Management environment, you can propagate data:

- From one Information/Management database to other Information/Management databases. Use the Automatic Log Save Facility.
- From Information/Management to DB2 databases. Use the DB2 Extract Facility.
- From Information/Management to other Information/Management databases and to DB2 databases. Use the Automatic Log Save Facility and the DB2 Extract Facility.

The Automatic Log Save Facility provides a means to propagate changes in a production Information/Management database to other Information/Management databases. The DB2 Extract Facility provides a means to propagate changes in a production Information/Management database to DB2 databases. In the Information/Management environment, the production database is referred to as the send database. The send database must be running with a structured description log data set (SDLDS), which captures a copy of all changes made in the database. Changes in the send database result from successfully creating, updating, and deleting records.
An Information/Management database that receives propagated data from a send database is referred to as a receive database. A receive database cannot be running with an SDLDS. A DB2 database that receives propagated data from a send database is referred to as the DB2 database. The process of propagation is transparent to the end user working on any of the databases: send, receive, or DB2.

Data propagation by either the Automatic Log Save Facility or the DB2 Extract Facility relies on a special type of Information/Management record, the LOGSAVE record. The LOGSAVE record provides a place for you to store your processing options, such as how often to propagate changes and whether to synchronize send and receive processing. See “Understanding the LOGSAVE Record” on page 11-15 for details on the LOGSAVE record.

The DB2 Extract Facility also uses relational data mapping tables (RDMTs) to determine how to map your Information/Management records to the DB2 database. You create RDMTs with BLGUT8 and store them in a data set that is in your BLGRFT concatenation.

Both data propagation facilities use batch facilities and network job entry (NJE). To initiate data propagation, you submit a batch job. The batch job offloads the SDLDS to a sequential data set, processes the sequential data set, and includes the processed data as in-stream data in another batch job that is routed to the MVS image where the receive database or the DB2 database resides. If the receive batch job is successful, it submits a batch job for routing to the MVS image where the send database resides to start the next send operation. So the receive is dependent upon the success of the send, and the next send is dependent upon the success of the receive. If the send encounters an error, the receive job is not submitted; if the receive job encounters an error, the next send job is not submitted.

Figure 7 illustrates data propagation to an Information/Management receive database and summarizes the steps involved in setting up the process. Figure 8 on page 11-3 illustrates data propagation to a DB2 database and summarizes the steps involved in setting up the process. Figure 9 on page 11-4 illustrates data propagation to both an Information/Management receive database and a DB2 database.

Figure 7. Data Propagation from Information/Management to Information/Management
Propagating Data

Understanding Data Propagation

Figure 8. Data Propagation from Information/Management to DB2
Mode of Propagation

Both the Automatic Log Save Facility and the DB2 Extract Facility support asynchronous propagation. This means that updates are applied at a later time and not within the same unit-of-work as the create, update, and delete functions on the send database. The amount of time that elapses between an update on the send database and the propagation of the update is partially determined by the value you specify for the **Time interval in minutes** field in the LOGSAVE record. (Other factors that determine the time interval are discussed in “Frequency of Data Propagation” on page 11-5.) Because a time interval elapses between an update on the send database and the update’s propagation, the propagated data is suitable for decision support applications, such as queries and reports.

You can propagate Information/Management data to DB2 running on the same MVS image as the send database or on a different image. You can also propagate Information/Management data to Information/Management running on the same MVS image as the send database or on a different image.

At this time, there is no utility for propagating existing Information/Management records to DB2. Because DB2 Extract uses the SDLDS to capture changes on the
send database, only those Information/Management records that are created or modified after the DB2 Extract Facility is started are propagated.

**Choosing the Data to Be Propagated**

For the Automatic Log Save Facility, you can choose to propagate either all of the records from the SDLDS or only those records that contain a particular prefix and data combination. Propagating only those records that contain a particular prefix and data combination is referred to as **filtering**. Filtering is done by the Automatic Log Save Receive function. You specify the filter prefix and filter data on the LOGSAVE record for the receive database.

For the DB2 Extract Facility, you choose the records and fields to propagate when you create the RDMTs that map your Information/Management records to DB2 tables. You use BLGUT8 to build an RDMT for each type of Information/Management record you want to propagate to DB2. In the input statements to BLGUT8, you map the fields in the record to DB2 tables and columns. You store the RDMTs in a data set that is in your BLGRFT concatenation and list their names in the LOGSAVE record.

**Ensuring the Integrity of the Propagated Data**

Propagated data is meant to be a copy of production data. Therefore, you want to ensure that all changes made to the production data are propagated to the receiving system in the same order in which they were made on the production system. Synchronization enables you to do this. See “Maintaining Synchronization” on page 11-22 for a discussion of synchronization and how it is maintained.

After the production data is propagated to the receiving database, use the propagated data as read-only to preserve its integrity.

**Frequency of Data Propagation**

Several factors determine the frequency of data propagation. You specify one of the factors in the **Time interval in minutes** field of the LOGSAVE record. Other factors that affect the frequency are how long it takes to:

- Offload the SDLDS
- Build the receive job
- Route and process the receive job
- Route the job to start the next send

Consider the following factors as well when deciding on a value for the time interval:

- Your network configuration and the amount of traffic it carries
- Whether you are synchronizing the Automatic Log Save Send and Receive functions or the DB2 Extract Send and Update Utilities
- The information about Information/Management enqueues in “Performance and Environmental Considerations” on page 11-14.
Using Propagated Data

Once propagation is established, you can develop programs that use the propagated Information/Management or DB2 data for reports and decision support purposes. Figure 10 and Figure 11 on page 11-7 show these programs as read-only to preserve the integrity of the Information/Management receive and DB2 databases. Any updates made to these databases are not propagated back to the send database.

Figure 10. Access to Data Propagated to Information/Management. Application A could be, for example, an interactive user; Application B could be an RFT.
Figure 11. Access to Data Propagated to DB2. Application A could be, for example, an application program; Application B could be a QMF user.

**Automatic Log Save Facility**

The Automatic Log Save Facility propagates Information/Management data to other Information/Management databases. It consists of the Automatic Log Save Send Function and the Automatic Log Save Receive Function. This section gives a brief overview of the two functions. For information about using the Automatic Log Save Facility, see Propagating to an Information/Management Database. Figure 12 on page 11-9 illustrates the major components and data flows of the Automatic Log Save Facility.

**Automatic Log Save Send Function**

The Automatic Log Save Send Function offloads the SDLDS to a sequential data set, referred to as the *send data set*, and includes the data from the send data set in-stream as part of a job to perform the Automatic Log Save Receive Function. After successful processing, the SDLDS is empty.

The Automatic Log Save Send Function reads the LOGSAVE record from the send database to determine the following:

- Send data set characteristics
- Whether to synchronize the send and receive functions
You start the Automatic Log Save Send Function by submitting JCL BLMSASD. This JCL starts Information/Management in batch and starts TSP BLGDUMP1, which calls user exits BLGUT4WT and BLGUT4EX.

See “Understanding Automatic Log Save Send Processing” on page 12-3 for details on the Automatic Log Save Send Function.

**Automatic Log Save Receive Function**

The Automatic Log Save Receive Function receives the send data set from an Information/Management user running the Automatic Log Save Send Function into the *receive data set* and adds the records contained in the receive data set to the receive database.

The Automatic Log Save Receive Function reads the LOGSAVE record to determine the following:

- Receive data set characteristics
- Whether to synchronize the send and receive functions
- How long to wait before processing the receive data set
- Which records in the receive data set to filter into the receive database.

The Automatic Log Save Receive Function is started by JCL BLMSARV, which calls user exits BLGUT3WT and BLGUT3EX. BLMSARV is built and submitted by JCL BLMSASD or BLMSASDA.

DB2 Extract Facility

The DB2 Extract Facility propagates Information/Management data to a DB2 database. It consists of the DB2 Extract Send Utility and the DB2 Update Utility. This section gives a brief overview of the two utilities. For information about using the DB2 Extract Facility, see Propagating to a DB2 Database. Figure 13 on page 11-11 illustrates the major components and data flows of the DB2 Extract Facility.

DB2 Extract Send Utility

The DB2 Extract Send Utility generates SQL statements to reflect activity in the Information/Management send database. It uses the send data set created by user exit BLGUT4EX and RDMTs created by BLGUT8 to construct the SQL statements. It writes the SQL statements to the SQL data set.
The DB2 Extract Send Utility reads the LOGSAVE record to determine the following:

- Send data set name
- Whether to synchronize the send and update utilities
- How long to wait before offloading the SDLDS
- The authorization ID for the DB2 tables
- The names of the RDMTs to use
- SQL data set characteristics.

You start the DB2 Send Utility by submitting JCL BLMSDSD. This JCL starts Information/Management in batch and starts TSP BLGTDBXM, which calls user exits BLGUT4WT, BLGUT4EX, and BLMSSGEN.

See “Understanding DB2 Extract Send Processing” on page 13-2 for details on the DB2 Extract Send Utility.

**DB2 Update Utility**

The DB2 Update Utility starts DSNTEP2, a PL/I program supplied by DB2, to update the DB2 database with the SQL statements sent by the DB2 Send Utility.

The DB2 Update Utility is started by JCL BLMSRCV. BLMSRCV is built and submitted by BLMSSND.

See “Understanding DB2 Update Processing” on page 13-4 for details on the DB2 Update Utility.
Fig. 13. DB2 Extract Facility: Major Components and Data Flows
Considerations for Using Both Data Propagation Facilities

You can use both data propagation facilities, the Automatic Log Save Facility and the DB2 Extract Facility, simultaneously in one send database. You can do this because, as Figure 14 on page 11-13 illustrates, both facilities use the same:

- LOGSAVE record
- Send data set created by BLGUT4EX

In other words, both data propagation facilities use a piece of the Automatic Log Save Send Function: the piece that offloads the SDLDS.

A successful Automatic Log Save Receive Function starts the next Automatic Log Save Send Function. When you are using both facilities, if the Automatic Log Save Receive Function fails, both facilities stop. However, if DB2 Update Utility fails but the Automatic Log Save Receive Function succeeds, the Automatic Log Save Facility continues. Therefore, another Automatic Log Save Send Function occurs, and another DB2 Send Utility occurs as well. In this situation, the DB2 Send Utility does not stop until the Automatic Log Save Facility encounters an error or you disable the LOGSAVE record.

To ensure the integrity of your DB2 data when you are using both facilities simultaneously, develop, test, and implement procedures for:

- Monitoring the operation of both facilities
- Recovering from errors

Error recovery is discussed in “Propagating to an Information/Management Database” on page 12-1 and “Propagating to a DB2 Database” on page 13-1.
Considerations for Using Both Data Propagation Facilities

Figure 14. Data Flows When Propagating Data to Both Information/Management and DB2
Performance and Environmental Considerations

Before you set up either the Automatic Log Save Facility or the DB2 Extract Facility, consider the following points:

- The Automatic Log Save Receive Function performs better if the receive database is not shared.
- To prevent duplicate records on the receive database, do not create any records on the receive database.
- BLGUT3EX obtains an exclusive enqueue on the receive database when adding records to the database. The value of the COGENQ parameter for the session-parameters member determines how long the enqueue is held. The default value for COGENQ is 10. If you increase the value for COGENQ, you increase the number of records BLGUT3EX can process each time it obtains an enqueue at the expense of other users. Refer to the Planning and Installation Guide and Reference for more information about session-parameters members and the COGENQ parameter.
- BLGUT4EX obtains an exclusive enqueue on the send database when it offloads the SDLDS to the send data set. Therefore, no records can be filed in the send database while the SDLDS is being offloaded. The RECS= and the %FULL= parameters on user exit BLGUT4EX enable you to control how long the exclusive enqueue is held. As shipped, TSPs BLGDUMP1 and BLGTDBXM specify a parameter of RECS=20. This means that BLGUT4EX obtains an exclusive enqueue on the SDLDS, processes 20 physical records, then releases the enqueue so that other users can access the SDLDS. BLGUT4EX then re-obtains the enqueue and processes 20 more physical records from the SDLDS, and so on. If you increase the number of records BLGUT4EX can process each time it obtains an exclusive enqueue, you increase the speed of BLGUT4EX at the expense of other users. If you decrease the number of records BLGUT4EX can process each time it obtains an exclusive enqueue, you decrease the impact that BLGUT4EX has on other users at the expense of slowing down BLGUT4EX. See Data Propagation User Exits for more information on the parameters.
- The DB2 Update Utility submits dynamic SQL statements with a COMMIT statement after the DELETE, INSERT, and UPDATE statements for each RNID processed. Refer to the IBM Database 2 Version 2 Administration Guide for information on the size and type of locks DB2 obtains.
- Batch jobs are used to start the offloading of the SDLDS for data propagation and to transfer, receive, and load the offloaded data. A successful offload causes the receive batch job to be built, submitted, and routed to the receiving system. If the receive batch job successfully loads the data, it submits a batch job to start another offload. Batch jobs can be delayed or can fail, sometimes without notifying you. Therefore, you need to implement procedures for:
  - Checking the status of the batch jobs
  - Checking the status messages from the batch jobs
Checking the output from the batch jobs
Error recovery
Restarting data propagation

If you want to use a different approach to data propagation, you can tailor the JCL that is provided in the sample library (SBLMSAMP). For example, you might want the Automatic Log Save Facility to stop after one successful receive operation. To do that, tailor the Automatic Log Save Facility as described in “Disabling Automatically Starting the Next Send” on page 12-33. Or you might want to use the Automatic Log Save Facility to maintain a backup copy of your SDLDS in a GDG but not to propagate your data. To do that, tailor the Automatic Log Save Facility as described in “Using the Automatic Log Save Facility for Automatic Backup” on page 12-33. In addition to those changes, you can, of course, make others, such as change the JCL to use a locally available method of data transfer instead of NJE.

Understanding the LOGSAVE Record

The LOGSAVE record is an Information/Management record that contains data used by both the DB2 Extract Facility and the Automatic Log Save Facility.

To create, update, and delete a LOGSAVE record, you must have database administrator authority; to display or print the record, requires no authority.

An Information/Management database can contain only one LOGSAVE record. The Automatic Log Save Facility uses the information on panels BLG0Z500, Automatic Log Save Data Set Information Entry, and BLG0Z510, Automatic Log Save Synchronize and Filter Entry. The DB2 Extract Facility uses information on those panels plus information on panels BLG0Z530, DB2 Extract Information Entry, and BLGLSQMP, Relational Data Mapping Table Entry.

Panels BLG0Z500, BLG0Z510, BLG0ZU50, BLG0Z530, and BLGLSQMP are provided to assist you in creating a LOGSAVE record. To prevent errors, do not change the logic of these panels or any of the other panels that are involved with creating or updating the LOGSAVE record.

The following descriptions of the LOGSAVE record’s fields are organized by panel.
**Fields on Panel BLG0Z500**

On this panel, you define the name and DCB characteristics for the send or receive data set. If the **Database type** field contains *send*, the data set defined is the send data set. If the **Database type** field contains *receive*, the data set defined is the receive data set. Both the Automatic Log Save and the DB2 Extract Facilities use the send data set. Only the Automatic Log Save Facility uses the receive data set.

<table>
<thead>
<tr>
<th>BLG0Z500 AUTOMATIC LOG SAVE DATA SET INFORMATION ENTRY RNID: LOGSAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter information used to send data from the SDLDS or receive the transmitted SDLDS data.</td>
</tr>
</tbody>
</table>

Send or Receive data set characteristics:

1. Data set name.........<R> ____________________________________________
2. Logical record length.......<R> 82/zerodot2_
3. Block size..................<R> 82/zerodot6_
4. Primary allocation blocks...<R> 200_
5. Secondary allocation blocks..<R> 100_
6. Unit type...................<R> SYSDA___
7. ABEND disposition...........<R> KEEP___
8. Time interval in minutes....<R> 10__
9. Database type...............<R> _______

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

---

**Database type**

Specifies whether the record is for a send database or a receive database. You can have only one LOGSAVE record in an Information/Management database. For the Automatic Log Save Facility, you create the LOGSAVE record on a send database and copy it to the receive database where you update it. Acceptable values are *send* and *receive*.

**Data set name**

Specifies the name of a sequential data set.

- For a send database, this data set is the one that the SDLDS is offloaded to. This data set is created by user exit BLGUT4EX using the DCB characteristics specified on this panel.
- For a receive database, this data set is the one that the send data set is received as by JCL BLMSARV.

**Logical record length**

Specifies the length of the logical records in the sequential data set. If you set this field to 0, it defaults to 8202. Refer to the *Planning and Installation Guide and Reference* for information on how to determine this value.
Block size
Specifies the block size of the data set. This value must be valid for the specified logical record length. If you set this field to 0, it defaults to 8206. Refer to the Planning and Installation Guide and Reference for information on how to determine this value.

Primary allocation blocks
Specifies how many blocks to allocate for the data set. This value is used only by the Automatic Log Save Send function. The number of blocks to allocate depends on the volume of records written to the SDLDS and how often you do send processing. The greater the volume of data written to the SDLDS and the longer the time interval between send processing operations, the larger the data set needs to be.

Secondary allocation blocks
Specifies how many blocks to allocate if the send data set runs out of primary allocation space. This value is used only by the Automatic Log Save Send function.

Unit type
Specifies the unit on which the send data set is to be allocated. This value is used only by the Automatic Log Save Send function.

ABEND disposition
Specifies what to do with the data set created by the send function or allocated by the receive function if either ends abnormally. Valid values are:

DELETE Specifies that the data set is not needed, and its space on the volume is to be released if the function ends abnormally.

KEEP Specifies that the data set is to be kept on the volume if the function ends abnormally.

CATLG Specifies that an entry pointing to the data set is to be placed in the system or user catalog when the data set is allocated, and the data set is to be kept if the function ends abnormally. Any missing index levels are created.

UNCATLG Specifies that the entry pointing to the data set in the system or user catalog, and any unneeded indexes (with the exception of the highest level), are to be deleted.

Time interval in minutes
Specifies how many minutes to wait before a send operation or a receive operation. When the send TSP (BLGDUMP1 or BLGTDBXM) or the receive TSP (BLGLOAD1) is started, the TSP reads the LOGSAVE record, then waits for the specified time interval (user exit BLGUT4WT for send or BLGUT3WT for receive). After the time interval elapses, control returns to the TSP, and the log or receive data set is processed. Consider your network configuration, how much traffic it carries, and the amount
Understanding the LOGSAVE Record

of time it will take to do an Automatic Log Save Receive or DB2 Update operation when you choose a value for the time interval.

If you use both the Automatic Log Save Facility and the DB2 Extract Facility, both facilities operate on the same time interval.

Fields on Panel BLG0Z510

On this panel, you specify whether to synchronize the Automatic Log Save Send and Receive functions or the DB2 Extract Send and Update utilities. If you use both facilities, the choice you make for synchronization affects both of them.

On this panel, you can also specify filter information for the Automatic Log Save Receive function. The LOGSAVE record is for the Automatic Log Save Receive function if the Database type field on panel BLG0Z500 contains receive.

<table>
<thead>
<tr>
<th>BLG0Z510 AUTOMATIC LOG SAVE Synchronize and Filter Entry RNID: LOGSAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter information used by the Automatic Log Save Send and Receive functions and by the DB2 Extract Send and Load functions.</td>
</tr>
<tr>
<td>1. Synchronize Send and Receive?.....&lt;R&gt; YES</td>
</tr>
<tr>
<td>Automatic Log Save Receive filter information:</td>
</tr>
<tr>
<td>NOTE: On Automatic Log Save Receive, any records that have the filter prefix and filter data combination are put into the Information/Management database. You must supply both a filter prefix and filter data to enable the filtering function.</td>
</tr>
<tr>
<td>2. Receive filter prefix................. _____</td>
</tr>
<tr>
<td>Receive filter data:</td>
</tr>
<tr>
<td>3. ____________________________________________</td>
</tr>
</tbody>
</table>

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

Synchronize Send and Receive?

Specifies whether to synchronize the Automatic Log Save Send and Receive functions or the DB2 Extract Send and Update Utilities.

Receive filter prefix

Specifies the prefix that is in the records you want to receive. This field is used only by the Automatic Log Save Receive Function. This field and the Receive filter data field identify records for receive processing. This field is used only when you specify receive as the database type and you specify a value for the Receive filter data field.
If you have a value in this field but no value in the **Receive filter data** field, this field is ignored.

**Receive filter data**

Specifies the data that is associated with the prefix in the records you want to receive. Together, this field and the **Receive filter prefix** field identify records for receive processing. This field is used only by the Automatic Log Save Receive Function. This field is used only when you specify **receive** as the database type and you specify a value for the **Receive filter prefix** field. If you have a value in this field but no value in the **Receive filter prefix** field, this field is ignored. The data in this field is converted to uppercase before using it to perform filter processing on Log Save **receive**.

### Fields on Panel BLG0ZU50

On this panel, you can review a summary of the information in the LOGSAVE record and quickly change the status of the record.

<table>
<thead>
<tr>
<th>BLG0ZU50</th>
<th>AUTOMATIC LOG SAVE RECORD SUMMARY</th>
<th>RNID: LOGSAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set name.</td>
<td>Date entered.</td>
<td>Database type.</td>
</tr>
<tr>
<td>Receive filter prefix.</td>
<td>Date last altered.</td>
<td>Receive filter data:</td>
</tr>
<tr>
<td>SQL data set name.</td>
<td>Status: DISABLED</td>
<td></td>
</tr>
</tbody>
</table>

Select one of the choices, or type **END** to save or **CANCEL** to discard changes.
1. Automatic Log Save data set information.
2. Automatic Log Save synchronize and filter information.
3. DB2 Extract information.
4. Toggle record status (ENABLED/DISABLED).

### Status

Specifies whether the LOGSAVE record is *enabled* or *disabled*. The status must be enabled for Automatic Log Save processing or DB2 Extract processing, or both, to occur. Select **Toggle record status** to set the status to *enabled* or *disabled*. 
Fields on Panel BLG0Z530

On this panel, you define the name and DCB characteristics for the SQL data set. The DB2 Extract Send Utility writes SQL statements to the SQL data set.

<table>
<thead>
<tr>
<th>BLG0Z530</th>
<th>DB2 EXTRACT INFORMATION ENTRY</th>
<th>RNID: LOGSAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter information used by the DB2 Extract Send function.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL data set characteristics:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Data set name..........&lt;R&gt; ____________________________________________</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical record length........... 80</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Block size..................&lt;R&gt; 5600_</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Primary allocation blocks....&lt;R&gt; 200_</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Secondary allocation blocks...&lt;R&gt; 100_</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Unit type.......................&lt;R&gt; SYSDA_</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. ABEND disposition.............&lt;R&gt; KEEP_</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Authorization ID............. &lt;R&gt; ________</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. Relational Data Mapping Tables</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Data set name
Specifies the name of the data set the DB2 Extract Send Utility is to allocate and use when generating SQL statements. This name must match the name you specify in the JCL member BLMSSND that processes the data set.

Logical record length
Specifies the length of the logical records in the SQL data set. This field is a display-only field and is set to 80.

Block size
Specifies the block size of the data set. This value must be valid for the logical record length. The SQL data set has a record format of fixed blocked.

Primary allocation blocks
Specifies how many blocks to allocate for the data set. Consider these factors to determine the size:
- The volume of data written to the SDLDS
- The time interval between send processing operations
- How many types of records you are generating SQL statements for
- How many fields you have mapped from those records.
The greater those factors are, the larger the data set needs to be. Consider, also, that two SQL statements (DELETE and INSERT) are generated for each Information/Management record found in the SDLDS as a result of a create or update function. (For each record found in the SDLDS as a result of a delete function, only a DELETE statement is generated.) If the Information/Management record maps to several DB2 tables, a pair of SQL statements (DELETE and INSERT) is generated for each table.

**Secondary allocation blocks**

Specifies how many blocks to allocate if the SQL data set runs out of primary allocation space.

**Unit type**

Specifies the unit on which the SQL data set is to be allocated.

**ABEND disposition**

Specifies what to do with the SQL data set created by the DB2 Extract Send Utility if it ends abnormally. Valid values are:

- **DELETE**
  Specifies that the data set is not needed and its space on the volume is to be released if the function ends abnormally.

- **KEEP**
  Specifies that the data set is to be kept on the volume if the function ends abnormally.

- **CATLG**
  Specifies that an entry pointing to the data set is to be placed in the system or user catalog when the data set is allocated, and the data set is to be kept if the function ends abnormally. Any missing index levels are created.

- **UNCATLG**
  Specifies that the entry pointing to the data set in the system or user catalog, and any unneeded indexes (with the exception of the highest level), are to be deleted.

**Authorization ID**

Specifies the authorization ID that is used for all DB2 tables referenced by the generated SQL statements in the SQL data set.

**Relational Data Mapping Tables**

This selection takes you to panel BLGLSQMP where you specify all the RDMTs you want to use for mapping Information/Management management records and fields to DB2 tables and columns. The RDMTs are built by BLGUT8 and are stored in a data set in your BLGRFT concatenation. List an RDMT name for each type of Information/Management record you want propagated to DB2.
Maintaining Synchronization

Under normal conditions, each send data set or SQL data set is received and loaded before the next send occurs. However, if you start a second send before the previous send is loaded (because of an error or data loss), a synchronization error can occur. A synchronization error indicates that the previous send data set or SQL data set was not successfully loaded.

Synchronization monitors the order in which the send data sets are loaded into the receive database and the SQL data sets update the DB2 database. A synchronization error notifies you that data is being loaded or updated out of order.

Synchronization is maintained by sequence numbers stored in the following places:

Send database’s SDIDS

The SDIDS record with the key of LAST_SDLDS_SENT contains the sequence number. This sequence number is incremented by 1, then written in the send data set. The first time BLGUT4EX offloads the SDLDS, this key does not exist. BLGUT4EX initializes the sequence number to 1 and writes the record in the send data set. After BLGUT4EX successfully offloads the SDLDS, the record is stored in the SDIDS. The next time BLGUT4EX runs, the sequence number is read from the SDIDS, incremented by 1, and that value (2) is put in the send data set.

You can use the GLOSSARY command to determine the value stored in this SDIDS record. Issue the IRC glossary,up max, find last from the Primary Options Menu to display the value. If your database is partitioned, you need a privilege class with master or universal access authority to see the LAST_SDLDS_SENT information in the glossary. Refer to the User’s Guide for more information about the GLOSSARY command.

Receive database’s SDIDS

The SDIDS record with the key of LAST_SDLDS_SENT contains the sequence number. This sequence number is 1 less than the sequence number the receive database expects to find in the next receive data set. (When the send data set is received by the Automatic Log Save Receive function, it becomes the receive data set.) The first time BLGUT3EX processes a receive data set, this key does not exist; BLGUT3EX initializes the sequence number to 1. After BLGUT3EX successfully receives and processes the receive data set, the record is stored in the SDIDS. The next time BLGUT3EX runs, the sequence number is read from the SDIDS, incremented by 1, and that value (2) is the sequence number that is expected to be in the receive data set.

You can use the GLOSSARY command to determine the value stored in this SDIDS record. Issue the IRC glossary,up max, find last from the Primary Options Menu to display the value. If your database is partitioned, you need a privilege class with master or universal access authority to see the LAST_SDLDS_SENT information in the glossary. Refer to the User’s Guide for more information about the GLOSSARY command.
**DB2 table SYNCH_TABLE**

You create a one-column table named `authorization_id.SYNCH_TABLE` in the DB2 database. The column is named `SYNCH_NUMBER`. Then you prime this column with the sequence number that is in the send database’s SDIDS. If the SDIDS of the send database does not contain a record with a key of LAST_SDLDS_SENT, prime this column with 0. Based on the sequence number in the send data set, the DB2 Extract Send Utility generates SQL statements that check the value in this table. (This table must never contain more than one row.)

**The SQL data set**

The first SQL statement in this data set does the synchronization checking. If the check fails, the DB2 Update Utility stops. If the check is successful, the updating continues and the last SQL statement in the SQL data set updates the value in `SYNCH_NUMBER` by 1.

For example, the first time BLGUT4EX processes, the send database’s SDIDS does not contain a sequence number, so the number that is written in the send data set is 1. If your value for `authorization_id` is BLM2DBX, the DB2 Extract Send Utility generates the following SQL statements, which appear as the first and last statements in the SQL data set:

```sql
UPDATE BLM2DBX.SYNCH_TABLE
SET SYNCH_NUMBER = SYNCH_NUMBER/(SYNCH_NUMBER - SYNCH_NUMBER)
WHERE SYNCH_NUMBER ~= 00000000;
;
UPDATE BLM2DBX.SYNCH_TABLE
SET SYNCH_NUMBER = 0000000001
WHERE SYNCH_NUMBER ~= 00000000;
```

If `SYNCH_NUMBER` does not contain a value of 0, the first statement results in message DSNT408I SQLCODE = -802, a 0 divide error. The DB2 Update Utility stops with a return code of 0008 and sends a failure message.

If `SYNCH_NUMBER` contains a value of 0, the DB2 Update Utility sends a success message and update processing continues. If none of the subsequent SQL statements ends abnormally, the last statement updates `SYNCH_NUMBER` to 1.

**The send and receive data set**

The first record of the send and receive data sets contain the LAST_SDLDS_SENT record in hex. (When the send data set is received by the Automatic Log Save Receive function, it becomes the receive data set.) The sequence number in this record is equal to the value that was stored in the SDIDS record in the send database after the send data set was created. The sequence number in this record is also equal to the value in the SDIDS record in the receive database plus 1. For example, the first time BLGUT4EX processes, this record contains a value of 1. Depending on the length of the SDIDS key for the database, this record has a key that is either 18 or 34 bytes long, followed by 8 bytes of miscellaneous data, followed by 4 bytes that contain the sequence...
Maintaining Synchronization

For a 34-byte SDIDS key, the sequence number is in columns 43 through 46. See Figure 15 on page 11-24 for an illustration. For an 18-byte key, the sequence number is in columns 27 through 30.

Figure 15. 34-byte SDIDS Key. The sequence number in columns 43 through 46 is X'00D5', which converts to decimal 213.

If synchronization is lost while propagating data to an Information/Management receive database, receive and send processing stop and a message is sent to the user ID specified in JCL BLMSARV. See “Restoring Synchronization” on page 12-37 for more information about Automatic Log Save synchronization and how to restore it.

If synchronization is lost while propagating data to a DB2 database, the DB2 Extract Facility stops unless you are running both the DB2 Extract Facility and the Automatic Log Save Facility. If you are running both, only the DB2 Update Utility stops. A message is sent to the user ID specified in JCL BLMSRCV. See “Recovering from Errors” on page 13-38 for more information about DB2 Extract synchronization and how to restore it.
Propagating to an Information/Management Database

This chapter contains Product-sensitive Programming Interface and Associated Guidance Information.

This information is for the program administrator responsible for implementing the Automatic Log Save Facility. You must understand Terminal Simulator Panels (TSPs), the Panel Modification Facility (PMF), and job control language (JCL) to do the tasks described in this chapter. For information about TSPs, refer to the Terminal Simulator Guide and Reference. For information about PMF, refer to the Panel Modification Facility Guide. For information about JCL, refer to the following publications:

- MVS/ESA JCL Reference
- MVS/ESA JCL User’s Guide

You must also know how to use BLGUT1, the BLX-SP operator commands, and IDCAMS. For information about BLGUT1 and the BLX-SP operator commands, refer to the Operation and Maintenance Reference. For information about IDCAMS, refer to the following publications:

- DFSMS/MVS Access Method Services for the Integrated Catalog Facility
- DFSMS/MVS Macro Instructions for Data Sets
- DFSMS/MVS Using Data Sets

The Automatic Log Save Facility provides two Information/Management functions:

**Automatic Log Save Send**
This function periodically offloads the Information/Management SDLDS to a sequential data set. You can set up this function to automatically send the sequential data set to other Information/Management databases, thereby propagating your production Information/Management data.

**Automatic Log Save Receive**
This function is started automatically by a successful Automatic Log Save Send function. This function can run on the same Information/Management system as the send function, but with a different session-parameters member, or it can run on a different Information/Management system. You can set up the receive function
to receive all the records in the offloaded SDLDS or to receive all the records except those with a specified prefix and data combination. The records in the receive databases provide copies of your production data that you can use for reporting.

A successful receive function starts the next send function.

You can also set up the send function to only offload the SDLDS to a generation data group (GDG) to automatically back up your production database and empty your SDLDS.

Before you can use the Automatic Log Save Facility, you must define a LOGSAVE record on your Information/Management send database. To propagate data to a receive database, you must copy the record from the send database to the receive database and update it. An Information/Management database can have only one LOGSAVE record. Therefore, a database can be either a send database or a receive database; it cannot be both a send and a receive database.

**Implementing the Automatic Log Save Facility**

Before you implement Automatic Log Save processing, you must understand the process and perform the following tasks:

1. Read “Propagating Information/Management Data” on page 11-1 for a discussion of data propagation and a description of the LOGSAVE record.

2. Read this chapter for details of Automatic Log Save Send processing, Automatic Log Save Receive processing, and the implementation tasks.

3. Decide how your installation will use the Automatic Log Save Facility.
   - Do you want to propagate your production data to a receive database?
   - What user ID will run the send function?
   - What user ID will run the receive function?
   - Which session-parameters member will contain the send database?
   - Which session-parameters member will contain the receive database?
   - Which session-parameters member on the receive database will have read/write access for updating the LOGSAVE record?
   - How often do you want to offload the SDLDS of the send database?
   - Do you want to change how long an enqueue on the SDLDS is held for offloading?
   - Do you want to synchronize the Send and Receive functions?
   - What user ID will receive status messages?
   - What user ID will receive the job outputs?
   - Do you want to receive all the records you send?
   - Do you want to receive a specific subset?
Implementing the Automatic Log Save Facility

- How many backup levels of the GDG will you keep?
- What procedures will you use for backup and disaster recovery?
- How will the automatic offload of the SDLDS fit in with your current database recovery procedures?
- Do you want to disable automatically starting the next Automatic Log Save Send operation?
- Do you want to automatically offload the SDLDS to backup your production database without propagating the data to a receive database?
- Do you want to propagate Information/Management data to both DB2 and Information/Management? If so, how will your Automatic Log Save processing fit in with your DB2 Extract processing?
- How do you want to handle error recovery?

4. Develop and test your backup and disaster recovery procedures.
5. Develop procedures for monitoring the operation of the Automatic Log Save Facility.

Understanding Automatic Log Save Send Processing

The Automatic Log Save Send function reads the LOGSAVE record to obtain the following information:

- Sequential data set characteristics, including:
  - Data set name
  - Logical record length
  - Block size
  - Primary allocation blocks
  - Secondary allocation blocks
  - Unit type
  - ABEND disposition
- Database type
- How long to wait before offloading the SDLDS
- Whether to tag the offload data set with a sequence number so that send and receive processing can be synchronized
- Whether send processing is enabled or disabled.

The Automatic Log Save Send function is started by JCL BLMSASD. This JCL does the following tasks:

1. Starts TSP BLGDUMP1, which does the following tasks. See “Data Propagation User Exits” on page 14-1 for more information about the user exits.
   a. Calls user exit BLGUT4WT to:
      1) Read the LOGSAVE record.
Understanding Automatic Log Save Send Processing

2) Verify that the LOGSAVE record is for a send database.
3) Verify that the status of the LOGSAVE record is enabled.
4) Put selected information from the LOGSAVE record into the TSCA variable data area.
5) Wait the length of time specified in the LOGSAVE record; then return to the TSP.

b. Calls user exit BLGUT4EX to:

1) Read information from the TSCA variable data area.
2) Allocate the send data set.
3) If you are synchronizing the send and receive functions:
   a) Obtain the sequence number from the SDIDS.
   b) Add 1 to the sequence number.
   c) Write the sequence number as the first record in the send data set.
4) Move records from the SDLDS into the send data set.
5) If you are synchronizing the send and receive functions, store the sequence number in the SDIDS.

2. Sends a status message about the success or failure of the offload to the user ID specified in the JCL.

3. Copies the send data set to the GDG.

4. Builds and routes jobstream BLMSARV by concatenating the following pieces:
   a. JCL member BLMSARV
   b. The send data set as in-stream data (uses TSO TRANSMIT and the MVS utility IEBGENER)
   c. JCL member BLMSAL1

5. Send function is complete.

If you want to propagate data to DB2 as well as to Information/Management, see Propagating to a DB2 Database after you finish reading this one.

Understanding Automatic Log Save Receive Processing

The Automatic Log Save Receive function reads the LOGSAVE record to obtain the following information:

- Sequential data set characteristics, including:
  - Data set name
  - Logical record length
  - Block size
  - ABEND disposition
- Database type
- How long to wait before processing the receive data set
■ Whether to expect the send data set to be tagged with a sequence number so that send and receive processing can be synchronized
■ Whether receive processing is enabled or disabled
■ What filter information, if any, to use. If no filter information is specified, all the records are received, except the LOGSAVE record.

The Automatic Log Save Receive Function is started by BLMSARV, the JCL generated and submitted by the Automatic Log Save Send Function. This JCL does the following tasks:

1. Deletes any existing receive data set.
2. Processes the in-stream send data set and uses TSO RECEIVE to receive the data into the receive data set.
3. Starts TSP BLGLOAD1, which does the following tasks. See “Data Propagation User Exits” on page 14-1 for more information about the user exits.
   a. Calls user exit BLGUT3WT to:
      1) Read the LOGSAVE record.
      2) Verify that the LOGSAVE record is for a receive database.
      3) Verify that the status of the LOGSAVE record is enabled.
      4) Put selected information from the LOGSAVE record into the TSCA variable data area.
      5) Wait the length of time specified in the LOGSAVE record; then return to the TSP.
   b. Calls user exit BLGUT3EX to:
      1) Read information from the TSCA variable data area.
      2) If you are synchronizing the send and receive functions:
         a) Obtain the sequence number from the SDIDS.
         b) Add 1 to the sequence number.
         c) Verify that this sequence number matches the sequence number in the receive data set.
      3) Convert the records in the receive data set to Information/Management records for storage in the receive Information/Management database. The records that are converted for storage are those that contain the data specified by the Receive filter prefix and the Receive filter data fields in the LOGSAVE record. If those fields are blank, all the records in the receive data set are converted and stored, except for the LOGSAVE record.
      4) If you are synchronizing the send and receive functions, store the sequence number in the SDIDS.
4. Submits JCL member BLMSASDA. It is routed to the MVS image where the send database resides to begin the next send function.
5. Sends a status message about the success or failure of the onload to the user ID specified in the JCL.

6. Routes the job output to the user ID specified on the /*ROUTE PRINT card in the JCL.

**Setting Up the Automatic Log Save Send Function**

The tasks that you need to do to set up the Automatic Log Save Send Function depend upon whether you plan to propagate data to Information/Management only, DB2 only, or to both Information/Management and DB2.

To set up the Automatic Log Save Send Function for data propagation only to Information/Management, do the following tasks:

1. Verify that the production database is running with an SDLDS. Refer to the *Planning and Installation Guide and Reference* for more information.

2. Create a LOGSAVE record for a send database on the production database. See “Creating a LOGSAVE Record for a Send Database” on page 12-7 for information on this task.

3. Create a GDG for the send data set. See “Creating a GDG” on page 12-14 for more information on this task.

4. Prepare the JCL. See “Preparing the JCL” on page 12-15 for more information on this task.

5. Modify the BLGDUMP1 TSP that offloads the SDLDS if you want to change how long it can hold an enqueue on the SDLDS. (A default of RECS=20 is specified as a parameter to BLGUT4EX.) See “Modifying the BLGDUMP1 TSP” on page 12-20 for more information on this task.

6. Set up the Automatic Log Save Receive Function. See “Setting Up the Automatic Log Save Receive Function” on page 12-21 for more information on this task.

To set up the Automatic Log Save Send Function for data propagation only to DB2, do the following tasks:

1. Verify that the production database is running with an SDLDS. Refer to the *Planning and Installation Guide and Reference* for more information.

2. Create a LOGSAVE record for a send database on the production database. See “Creating a LOGSAVE Record for a Send Database” on page 12-7 for information on this task.

3. Do the setup for the DB2 Extract Facility in “Setting Up the DB2 Extract Facility” on page 13-5.

To set up the Automatic Log Save Send Function for data propagation to both Information/Management and DB2, do the following tasks:

1. Verify that the production database is running with an SDLDS. Refer to the *Planning and Installation Guide and Reference* for more information.
2. Create a LOGSAVE record for a send database on the production database. See “Creating a LOGSAVE Record for a Send Database” on page 12-7 for information on this task.

3. Prepare the JCL as described in “Preparing the JCL” on page 12-15. Make the additional changes described in “Propagating Data to Both Information/Management and DB2” on page 13-35.

4. Set up the Information/Management Receive database. See “Setting Up the Automatic Log Save Receive Function” on page 12-21 for more information on this task.

5. Do the setup for the DB2 Extract Facility in “Setting Up the DB2 Extract Facility” on page 13-5.

Creating a LOGSAVE Record for a Send Database

When you create a LOGSAVE record for a send database, you must specify information about the sequential data set, the time interval in minutes that the TSP will delay before offloading the SDLDS, and whether you want to synchronize the send and receive functions. Before you file the record, you must enable it. If you file the record with its status identified as disabled and then start the BLGDUMP1 TSP to offload the SDLDS, the TSP encounters an error.

To create the LOGSAVE record, select Entry from the Primary Options Menu for the System application. Type 5 and press Enter.

```
BLGOEN10 --- PRIMARY OPTIONS MENU --- APPLICATION: SYSTEM

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. DBADMIN..........Perform database administration.

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

Information/Management Version 1 Release 1
5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company, 1997
```

```
On BLG00010, the System Record Entry panel, select the kind of record you want to create.

For this exercise, select **Logsave**. Type 3 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.

+------------------------------- SELECT ITEM -----------------------------+

====>3
```

The next panel, the Automatic Log Save Data Set Information Entry panel, contains several required fields, some of which are primed. You can accept these values or change them. You must complete the **Data set name** field and the **Database type** field.

The amount of time you specify in the **Time interval in minutes** field is the amount of time the send TSP delays after being started by a successful receive operation. The actual time between offloads of the SDLDS is equal to the value in this field plus the time it takes to do each of the following tasks:

- Offload the SDLDS
- Build the receive JCL
- Route the receive JCL to the receive system
- Do the receive operation
- Route the next send JCL from the receive system to the send system
For this example, name the data set SEND.LOG; change the **Time interval in minutes** field to 60 minutes. Because you are creating the record for a send database, type this IRC:

```
1,send.log,8,60,9,send,end
```

and press Enter.
The Automatic Log Save Record Summary panel appears. The **Synchronize Send and Receive?** field defaults to YES. If you do not want to synchronize the send and receive functions, select **Automatic Log Save synchronize and filter information**. Type 2 and press Enter.

See “Ensuring the Integrity of the Propagated Data” on page 11-5 and “Maintaining Synchronization” on page 11-22 for information about synchronization.

<table>
<thead>
<tr>
<th>BLG0ZU50</th>
<th>AUTOMATIC LOG SAVE RECORD SUMMARY</th>
<th>RNID: LOGSAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set name.............. SEND.LOG</td>
<td>Date entered....... _____</td>
<td></td>
</tr>
<tr>
<td>Database type.............. SEND</td>
<td>Date last altered... _____</td>
<td></td>
</tr>
<tr>
<td>Time interval in minutes..... 0060</td>
<td>Time entered........ _____</td>
<td></td>
</tr>
<tr>
<td>Synchronize Send and Receive?. YES</td>
<td>Receive filter prefix....... _____</td>
<td></td>
</tr>
</tbody>
</table>

Receive filter data:

| SQL data set name.............. | Status: DISABLED |

Select one of the choices, or type END to save or CANCEL to discard changes.

1. Automatic Log Save data set information.
2. Automatic Log Save synchronize and filter information.
3. DB2 Extract information.
4. Toggle record status (ENABLED/DISABLED).


==> 2
The **Synchronize Send and Receive?** field is primed with YES. If you do not want the functions synchronized, change this field to NO. Filter information does not apply to a send database, so do not complete those fields. Type `end` and press Enter.

```
BLGOZ510 AUTOMATIC LOG SAVE SYNCHRONIZE AND FILTER ENTRY  RNID: LOGSAVE

Enter information used by the Automatic Log Save Send and Receive functions and by the DB2 Extract Send and Load functions.

1. Synchronize Send and Receive?.....<R> YES

Automatic Log Save Receive filter information:

   NOTE: On Automatic Log Save Receive, any records that have the filter prefix and filter data combination are put into the Information/Management database. You must supply both a filter prefix and filter data to enable the filtering function.

2. Receive filter prefix.............. _____

Receive filter data:
3. _______________________________________

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

   ===> end
```
The Automatic Log Save Record Summary panel appears. Notice that the **Status** is disabled. To enable the record for send processing, type 4 and press Enter.
Now the Status is enabled. Send processing does not actually occur, however, until you start the BLGDUMP1 TSP to perform the Automatic Log Save Send function.

You have created the record for a send database. To file the record, type 9 and press Enter.

BLG0EN10, the Primary Options Menu, appears with a message that the LOGSAVE record was stored successfully.
For future reference, you may want to print the LOGSAVE record.

Creating a GDG

This section pertains to creating the generation data group (GDG) base for the Automatic Log Save send data set. Job BLMSASD or BLMSASDA does the actual backup to a generation of the GDG you create in this procedure.

When you create the GDG base, you define how many generations of the GDG to keep by the value you specify on the LIMIT parameter.

To create a GDG for the send data set, modify and submit JCL member BLGDG1 from the sample library, SBLMSAMP. Using uppercase text:

1. Change the job card to your local standards.

2. On the BLGDG DD card, change userid.dsname.gdg to the name you want to use for the generation data group for the send data set.

3. On the DEFINE statement, change userid.dsname.gdg to the name you want to use for the generation data group for the send data set. This name must match the name you used on the BLGDG card.

4. Change any other options in the DEFINE statement at your discretion. For example, you may want to increase or decrease the value for LIMIT.

As shipped, the DCB information in BLGDG1 matches the values primed into the LOGSAVE record on the send database when the LOGSAVE record is created.
Preparing the JCL

Copy and edit each of the following JCL members from the sample library, SBLMSAMP. Do not submit these JCL members after you update them. One of them, BLMSASD, you submit later to start the Automatic Log Save Facility. See “Directions for Updating JCL” on page 12-16 for general directions on updating JCL.

**BLMSASD**  
Runs on the MVS image where the Information/Management send database resides. You submit this job to initiate the Automatic Log Save Facility. This job does the following tasks:

- Starts TSP BLGDUMP1 using Information/Management in batch mode.
- Sends a status message based on the success or failure of BLGDUMP1.
- Copies the Automatic Log Save send data set to a generation of a GDG.
- Prepares the send data set for inclusion in the receive job (BLMSARV).
- Builds the receive job (BLMSARV) containing the Automatic Log Save send data set as in-stream data.
- Routes job BLMSARV to the MVS image where the receive database resides for execution.

**BLMSASDA**  
Runs on the MVS image where the Information/Management send database resides. This job is submitted automatically by:

- BLMSARV following a successful receive operation
- BLMSASD or BLMSASDA if the send data set was empty.

Because this JCL can be submitted by BLMSARV, which runs on the MVS image where the receive database resides, you must copy this JCL to the MVS image where the receive database resides. You specify information to route this JCL to the MVS image where the send database resides.

This JCL member does the following tasks:

- Deletes the send data set created by the previous send operation.
- The tasks listed under BLMSASD.

**BLMSARV**  
A partial job that is combined with the in-stream send data set and BLMSAL1 to create the job stream that is routed to the MVS image where the receive database resides for execution. It begins a step that copies the in-stream records from BLMSASD or BLMSASDA to a temporary data set.
BLMSAL1  A partial job that is appended to BLMSARV after the in-stream records from the Automatic Log Save send data set. It does the following tasks:

- Completes the step that copies in-stream records to a temporary data set.
- Prepares the in-stream data and performs TSO RECEIVE to receive the data into the receive data set.
- Starts TSP BLGLOAD1 using Information/Management in batch mode.
- Sends a status message based on the success or failure of BLGLOAD1.
- If the receive is successful, submits job BLMSASDA, which is routed to the MVS image where the send database resides for execution.

BLMSASDE  Processes a previously created send data set that was not successfully received. This JCL must be manually submitted on the MVS image where the send database resides. It does the following tasks:

- Builds the job-stream (BLMSARV) containing the Automatic Log Save send data set as in-stream data.
- Routes job BLMSARV to the MVS image where the receive database resides for execution.

**Directions for Updating JCL**

1. Edit each member. Make your changes using uppercase text.
2. Follow the instructions in the prologue of each member and printed in this chapter, to find the specific job steps you need to update.
3. Verify the accuracy of each change you make against your LOGSAVE record, TSP, and other JCL members.
4. Be sure to change the job card to your local standards.
5. Save the updated member.

**Preparing BLMSASD**

1. Change the job card to your local standards.
2. Change the `node.userid` on the /*ROUTE PRINT card to the node and user ID to receive the job output.
3. In step OFFLD, change:
   a. The data set names for the STEPLIB, ISPPROF, ISPPLIB, ISPPLIB, ISPMLIB, ISPSLIB, BLGTRACE, ISPLOG, SYSPROC, and SYSTSPRT DDnames to correspond to the data set names at your installation.
b. The value for SESS(xx) to the session-parameters member of your send database.

4. In step GOODOFF, change:
   a. sample message to your message for a successful completion (return code = 0000) of the SDLDS offload.
   b. node.userid to the destination for the message.

5. In step FAILOFF, change:
   a. sample message to your message for an unsuccessful offload of the SDLDS (return code ≠ 0000).
   b. node.userid to the destination for the message.

6. In step SALESND, change:
   a. userid.dsname1 on the SYSUT1 and SYSUT2 DD names to the name of the send data set that is in the LOGSAVE record on the send database.
   b. userid.gdgname1 to the name of the GDG for the send data set.
   c. The DCB attributes for the GDG generation to match those used for the send data set.

7. In step XMITSEND, change userid.dsname1 to the name of the send data set that is in the LOGSAVE record on the send database.

8. In step SUBSEND, change the data set name to the name of your local data set that contains JCL BLMSASDA.

9. In step COPY1, change:
   a. The JCL data set name to the name of your local data set that contains JCL BLMSARV.
   b. The space DCB parameter on the SYSUT2 DD name to a value large enough to handle splitting each 80-byte record created from the transmit into two 80-byte records (the splitting occurs in step SPLIT).

10. In step COPY2, change the JCL data set name to the name of your local data set that contains BLMSAL1.

11. In step GOODESUB, change:
   a. sample message to your message for a successful submission of the receive JCL.
   b. node.userid to the destination for the message.

**Preparing BLMSASDA**

1. Change the job card to your local standards. This job is submitted on the MVS image where the send database resides. It is also submitted on the MVS image where the receive database resides and is routed to the MVS image where the send database resides for execution. You may need user and password information.
2. Change node on the /*ROUTE XEQ card to the node where the send database resides.

3. Change the node.userid on the /*ROUTE PRINT card to the node and user ID to receive the job output.

4. In step DELETE, change userid.dsname1 to the name of the send data set that is specified in the LOGSAVE record on the send database.

5. The rest of this JCL member is the same as BLMSASD beginning with step 3 under Preparing BLMSASD. Therefore, go to “Preparing BLMSASD” on page 12-16 and make the changes listed there beginning at step 3.

Copy this JCL to a data set on the MVS image where the receive database resides.

Preparing BLMSARV

1. Change the job card to your local standards. This job is routed to the MVS image where the receive database resides; you may need user and password information.

2. Change node on the /*ROUTE XEQ card to the node where the job is to run.

3. Change node.userid on the /*ROUTE PRINT to the node and user ID that is to receive the job output.

4. In step DELETE, change userid.dsname2 to the name of the receive data set that is specified in the LOGSAVE record on the receive database. This name must match the data set name in step RCVSND in JCL member BLMSAL1.

Preparing BLMSAL1

1. In DD card SYSUT2, which is part of the REBLK step started in BLMSARV, you may need to change the unit and space parameters.

2. In step RCVSND, change userid.dsname2 to the name of the receive data set that is specified in the LOGSAVE record on the receive database. This name must match the data set name in step DELETE step in JCL member BLMSARV.

3. In step JOIN, you may need to change the unit and space parameters.

4. In step ONLOAD, change:
   a. The data set names for the STEPLIB, ISPPROF, ISPPLIB, ISPMLIB, ISPSLIB, BLGTRACE, ISPLLIB, SYSPROC, and SYSTSPRT DD names to correspond to the data set names on the MVS image where the receive database resides.
   b. The value for SESS(yy) to the session-parameters member that has the receive database as database 5 with read/write access.

5. In step GOODON, change:
   a. sample message to your message for a successful completion (return code = 0000) of the onload.
   b. node.userid to the destination for the message.
6. In step FAILON, change:
   a. *sample message* to your message for an unsuccessful onload (return code ≠ 0000).
   b. *node.userid* to the destination for the message.

7. In step SUBSND, change the data set name to the name of your data set on the MVS image where the receive database resides that contains JCL **BLMSASDA**.

8. In step GOODSUB, change:
   a. *sample message* to your message for a successful submission of the next send JCL.
   b. *node.userid* to the destination for the message.
Modifying the BLGDUMP1 TSP

If you want to change the maximum number of records to offload each time an enqueue is obtained on the SDLDS, you can change the RECS= and %FULL= parameters in the control line that calls user exit BLGUT4EX. As shipped, the BLGDUMP1 TSP has RECS=20. See Data Propagation User Exits for more information about the parameters and the values they accept. The following example illustrates changing the RECS= parameter and adding the %FULL= parameter.

Using PMF, update the BLGDUMP1 TSP. Locate the control line for user exit BLGUT4EX. Update that line of the TSP.

On the Control Line Summary panel, select Data field specification. Type 1 and press Enter.

---

BLMBCU9L CONTROL LINE SUMMARY PANEL: BLGDUMP1

<table>
<thead>
<tr>
<th>Label name............</th>
<th>S-word index............</th>
<th>0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSQA field name........</td>
<td>Structured word........</td>
<td></td>
</tr>
<tr>
<td>Apply not logic........</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Get variable data......</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Panel name.............</td>
<td>Prefix word............</td>
<td>0000</td>
</tr>
<tr>
<td>Find string anywhere...</td>
<td>NO</td>
<td></td>
</tr>
<tr>
<td>Function exit...........</td>
<td>BLGUT4EX</td>
<td></td>
</tr>
</tbody>
</table>

Select one of the choices, or type END to save or CANCEL to discard changes.

1. Data field specification.
2. Flag field specification.

---

12-20 Version 1.1
You enter the parameters in the **Literal/Test data** field on the Data Field Specification panel. In this example, a value of 1000 for RECS= and a value of 50 for %FULL= have been entered.

<table>
<thead>
<tr>
<th>BLM8CU9P</th>
<th>DATA FIELD SPECIFICATION</th>
<th>PANEL: BLGDUMP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter 'USEREXIT' data fields; cursor placement or input line entry allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Function exit...........&lt;R&gt; BLGUT4EX</td>
<td>Structured word......</td>
<td></td>
</tr>
<tr>
<td>2. Structured word index....... 0000</td>
<td>Word acronym.........</td>
<td></td>
</tr>
<tr>
<td>3. Prefix index.................. 0000</td>
<td>Prefix..................</td>
<td></td>
</tr>
<tr>
<td>4. Label name...................</td>
<td>Validation...........</td>
<td></td>
</tr>
<tr>
<td>5. Panel name...................</td>
<td>New structured word....</td>
<td></td>
</tr>
<tr>
<td>6. Verify name..................</td>
<td>New word acronym......</td>
<td></td>
</tr>
<tr>
<td>7. TSCA field name...............</td>
<td>New prefix...........</td>
<td></td>
</tr>
<tr>
<td>8. New structured word index.. 0000</td>
<td>New validation........</td>
<td></td>
</tr>
<tr>
<td>9. New prefix index........... 0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10. User data..................</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. List index............... 0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13. Literal/Test data...........</td>
<td>RECS=1000 %FULL=50</td>
<td></td>
</tr>
<tr>
<td>14. New data....................</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

After you file the updated TSP, you must copy it from the write panel data set to the read panel data set so that it is available for use. Next, set up the receive database. After you do that, you are ready to start BLMSASD, the batch job that runs the TSP.

**Setting Up the Automatic Log Save Receive Function**

To set up the Automatic Log Save Receive Function, do the following tasks:

1. Define a receive database without an SDLDS in a session-parameters member. Define the receive database (SDDS and SDIDS) as a read-only database. You can do this in one of the following ways:
   - Define the SDDS and SDIDS for the receive database as database 5, and specify RDONLY=YES on the BLGCLDSN macro statements for the SDDS and SDIDS.
   - Define the SDDS and SDIDS for the receive database as database 4, 7, 8, or 9, so they cannot be written to. Specify a different SDDS and SDIDS for database 5.

   Use this session-parameters member for the applications that use the data in the receive database. Refer to the *Planning and Installation Guide and Reference* for information on defining session-parameters members.
2. Define a receive database without an SDLDS in a second session-parameters member. Define the receive database (SDDS and SDIDS) as database 5 with read/write access. This is the session-parameters member that the BLGLOAD1 TSP is run under. Therefore, do not create any records in this database. The program administrator needs read/write access to this database to update the LOGSAVE record. Refer to the Planning and Installation Guide and Reference for information on defining session-parameters members.

3. Copy the LOGSAVE record from the send database to the receive database. See “Copying the LOGSAVE Record into the Receive Database” for more information on this task.

4. Optionally, copy other records from the send database. See “Copying the LOGSAVE Record into the Receive Database” for more information on this task.

5. Update the LOGSAVE record. See “Updating the LOGSAVE Record for Receive” on page 12-24 for more information on this task.

6. Copy JCL BLMSASDA from the MVS image where the send database resides to the MVS image where the receive database resides. See “Copying the JCL” on page 12-32 for more information on this task.

Copying the LOGSAVE Record into the Receive Database

The LOGSAVE record in the Information/Management receive database must have the same VSAM key that the LOGSAVE record in the send database has. The following sections outline three methods you can use to copy the LOGSAVE record from the send database to the receive database.

Copying Entire Send Database without BLGUT1

This method uses IDCAMS REPRO to copy the entire SDDS and SDIDS from the send database to the receive database. You can use this method only if the receive database is empty.

1. On the send database, do the following tasks:
   a. Issue the BLX-SP FREE command on the send database’s SDDS and SDIDS so that no records can be created on the send database.
   b. REPRO the send database’s SDDS and SDIDS to a sequential data set using IDCAMS.
   c. Issue the BLX-SP REALLOC command on the send database’s SDDS and SDIDS.
   d. Do not start send processing, but you may start using your send database for other functions.

2. On the receive database, REPRO the SDDS and SDIDS from the send database into the receive database using IDCAMS. You may need to issue the BLX-SP FREE command if the receive database is allocated to a BLX-SP. Then, after the REPRO, issue the BLX-SP REALLOC command.
Propagating to a Database

Setting Up the Automatic Log Save Receive Function

Copying Entire Send Database with BLGUT1
This method uses IDCAMS REPRO to copy the entire SDDS from the send database to the receive database; then uses BLGUT1 to rebuild the SDIDS on the receive database. The receive database must be empty.

1. On the send database, do the following tasks:
   a. Issue the BLX-SP FREE command on the send database’s SDDS and SDIDS so that no records can be created on the send database.
   b. REPRO the send database’s SDDS to a sequential data set using IDCAMS.
   c. Issue the BLX-SP REALLOC command on the send database’s SDDS and SDIDS.
   d. Do not start send processing, but you may start using your send database for other functions.

2. On the receive database, do the following tasks:
   a. Using IDCAMS, REPRO the SDDS from the send database into the receive database. You may need to issue the BLX-SP FREE command if the receive database is allocated to a BLX-SP. Then, after the REPRO, issue the BLX-SP REALLOC command.
   b. Run BLGUT1 to build the receive database’s SDIDS. Refer to the Planning and Installation Guide and Reference for more information.

Copying Only the LOGSAVE Record from the Send Database
This method uses IDCAMS REPRO to copy just the LOGSAVE record from the SDDS of the send database to the SDDS of the receive database; then uses BLGUT1 to rebuild the SDIDS of the receive database. Use this method if you only want to put the LOGSAVE record from the send database in the receive database. The VSAM key for the LOGSAVE record must be available. (The LOGSAVE record on the receive database must have the same VSAM key as the LOGSAVE record on the send database.)

1. On the send database, do the following tasks:
   a. Issue the BLX-SP FREE command on the send database’s SDDS so that no records can be created on the send database.
   b. REPRO just the LOGSAVE record from the send database’s SDDS to a sequential data set using IDCAMS.
   c. Issue the BLX-SP REALLOC command on the send database’s SDDS.
   d. Do not start send processing, but you may start using your send database for other functions.

2. On the receive database, do the following tasks:
   a. Delete and redefine the receive database’s SDIDS.
      1) Issue the BLX-SP FREE command on the receive database’s SDIDS.
      2) Use IDCAMS to delete the receive database’s SDIDS.
      3) Use IDCAMS to define the receive database’s SDIDS.
b. Issue the BLX-SP FREE command on the receive database’s SDDS.
c. Using IDCAMS, REPRO just the LOGSAVE record from the send SDDS into the receive SDDS.
d. Issue the BLX-SP REALLOC command on the receive database’s SDDS and SDIDS.
e. Run BLGUT1 to build the receive database’s SDIDS.

Updating the LOGSAVE Record for Receive

When you update the LOGSAVE record that you copied from the send database, you must change the **Database type** field. You may also want to change the following fields:

- Data set name
- ABEND disposition
- Time interval in minutes

However, do not change the following fields:

- Logical record length
- Block size
- Synchronize Send and Receive

For receive processing, you may also specify values for the **Receive filter prefix** and **Receive filter data** fields. (The LOGSAVE record is automatically filtered out. So if you modify the LOGSAVE record on the send database, those changes are not propagated to the receive database.)

Before you file the record, you must be sure that the record is enabled. If you file the record with its status identified as disabled and then start the BLGLOAD1 TSP, no processing occurs.
To update the LOGSAVE record, start from the Primary Options Menu and type the following IRC:

```
update r logsave
```

and press Enter.
On BLG0ZU50, the Automatic Log Save Record Summary panel, select **Automatic Log Save data set information**. Type 1 and press Enter.

```
BLG0ZU50          AUTOMATIC LOG SAVE RECORD SUMMARY          RNID: LOGSAVE

Data set name.............. SEND.LOG
Database type................. SEND    Date entered........ 09/30/93
Time interval in minutes...... 0060    Time entered........ 17:10
Synchronize Send and Receive?. YES    Date last altered... 09/30/93

Receive filter prefix........ ______
Receive filter data:

SQL data set name........... ____________________________________________

Status......................... ENABLED

Select one of the choices, or type END to save or CANCEL to discard changes.
1. Automatic Log Save data set information.
2. Automatic Log Save synchronize and filter information.
3. DB2 Extract information.
4. Toggle record status (ENABLED/DISABLED).

===>
1
```
On the Automatic Log Save Data Set Information Entry panel, you must change the **Database type** field to **RECEIVE**. You must not change the **Logical record length** or the **Block size** fields.

Even though the Automatic Log Save Receive function does not use the **Primary allocation blocks**, **Secondary allocation blocks**, and **Unit type** fields, the **LOGSAVE** record requires values in the fields.

You may change any of the remaining fields, including the **Data set name** field. The data set name for the receive data set does not have to match the data set name for the send data set; however, the receive data set name does have to match the name you specified in the **DELETE** step in **BLMSARV** and in the **RCVSND** step in **BLMSAL1**. (See “Preparing **BLMSARV**” on page 12-18 and “Preparing **BLMSAL1**” on page 12-18.)

The time interval you specify on the **LOGSAVE** record for the receive database tells user exit **BLGUT3WT** how long to wait before **TSP BLGLOAD1** continues receive processing. If you want the records from the receive data set loaded into your receive database without a delay after **BLGLOAD1** **TSP** starts, change the **Time interval in minutes** field to 0. You can control how often the send and receive functions are performed by the time value you specify in the **LOGSAVE** record on the send database. For this example, name the data set **RCVE.LOG** and change the time interval to 0. Type this **IRC**:

```
1,rcve.log,8,0,9,receive,end
```

and press Enter.
The Automatic Log Save Record Summary appears. If you want to specify filtering information, select **Automatic Log Save synchronize and filter information**. Type:

2

and press Enter.

See “Choosing the Data to Be Propagated” on page 11-5 for more information about filtering.
Do not change the value in the **Synchronize Send and Receive?** field. The value in this field must be the same on both the send database and the receive database.

You do not want to receive all the records in the send data set. You want to receive only those records that have a location code of `CHAMB`. To do that, type:

```
2,locc/,3,chamb,end
```

and press Enter.
The Automatic Log Save Record Summary appears. The **Status** line says that the record is enabled.

You have finished updating the LOGSAVE record for a receive database. To file the record, type 9 and press Enter. Receive processing does not actually occur, however, until you start the BLGLOAD1 TSP to perform Automatic Log Save.
BLG0EN20, the Primary Options Menu, appears with a message that the LOGSAVE record was stored successfully.

For reference, print the LOGSAVE record.
Filtering Records
In this example, you specified LOCC/ in the **Receive filter prefix** field and CHAMB in the **Receive filter data** field. Therefore, only those records that contain the prefix and data combination LOCC/CHAMB are converted for storage in the receive Information/Management database. The remaining records are left in the receive data set (RCVE.LOG in this example) where they are deleted during the next receive processing operation.

The LOGSAVE record is automatically filtered out. So if you modify the LOGSAVE record on the send database, those changes are not propagated to the receive database.

Copying the JCL
Copy JCL member BLMSASDA from the MVS image where the send database resides and store it in a data set referenced by step SUBSND in BLMSAL1. (See “Preparing BLMSAL1” on page 12-18.) You need not change BLMSASDA after you copy it. Do not submit the copied BLMSASDA; BLMSARV submits it automatically after a successful receive operation to start the next offload of the SDLDS.

Starting Normal Send and Receive Processing
To start normal Send and Receive processing, start the Automatic Log Save Facility by submitting JCL BLMSASD on the MVS image where the send database resides. Before you submit JCL BLMSASD, verify that:

- The set up for your send database is complete.
- The set up for your receive database is complete.
- The status of both the send and receive LOGSAVE records is enabled.

Stopping Send and Receive Processing
To stop Automatic Log Save Send and Receive processing, disable the LOGSAVE record on the send database. You can use the following IRC from the Primary Options Menu to toggle the status:

`update r logsave,4,9`

When you stop the send function, the receive function also stops. If a send is currently in progress (that is, BLMSASDA or BLMSASD has been started and TSP BLGDUMP1 has called user exit BLGUT+WT and the exit is waiting), the current send finishes waiting. Then the current send offloads the SDLDS and routes BLMSARV for execution. The receive (BLMSARV) processes and submits BLMSASDA for execution on the MVS image where the send database resides. BLMSASDA fails in step OFFLD because of a TSP error indicating that the status of the LOGSAVE record is disabled.

Do not disable the receive LOGSAVE record. If you do, the send occurs, but the send data set will not be successfully received, and you will have to reprocess the send data set.
Using the Automatic Log Save Facility for Automatic Backup

You can use the Automatic Log Save Facility to automatically offload the SDLDS to maintain a backup copy of your production database without propagating your production data to a receive database. Using the Automatic Log Save Facility in this way does not require you to set up a receive database.

To set up the Automatic Log Save Facility for automatic backup only, do the following tasks:

1. Verify that the production database is running with an SDLDS. Refer to the *Planning and Installation Guide and Reference* for more information.

2. Create a LOGSAVE record for a send database on the production database. See “Creating a LOGSAVE Record for a Send Database” on page 12-7 for information on this task.

3. Create a GDG for the send data set. See “Creating a GDG” on page 12-14 for more information on this task.

4. Prepare JCL members BLMSASD and BLMSASDA as described in “Preparing BLMSASD” on page 12-16 and “Preparing BLMSASDA” on page 12-17. Make the following additional changes to both JCL members:
   a. Delete steps XMITSND, SPLIT, BLDJOB, SUBRCV, and GOODSUB.
   b. Modify step SUBSND:
      1) On the COND parameter, delete \((0,EQ,XMITSND),EVEN)\).
      2) On the IEFRDER DDname, change the data set name that identifies the JCL member to be submitted if you have copied BLMSASDA to another data set and member.

5. Modify the BLGDUMP1 TSP that offloads the SDLDS if you want to restrict how long it can hold an enqueue on the SDLDS. See “Modifying the BLGDUMP1 TSP” on page 12-20 for more information on this task.

6. Submit BLMSASD. When it completes successfully, it submits BLMSASDA.

Disabling Automatically Starting the Next Send

If you want to disable automatically starting the next Automatic Log Save Send Function when the Automatic Log Save Receive Function completes successfully, you can delete the step that submits the JCL to start the next send from the JCL for the Automatic Log Save Receive Function. Then, in the JCL you use to start the Automatic Log Save Send Function (either BLMSASDA or BLMSASD) you reference the modified JCL for the Automatic Log Save Receive.

To implement the Automatic Log Save Facility so that it does just one send and one receive, then stops, modify the JCL as follows:

1. Copy or update JCL BLMSAL1. Delete steps SUBSND and GOODSUB.
2. Copy or update JCL BLMSASDA (or BLMSASD if you want to manually delete the send data set after each send and receive) as follows:
   a. Delete step SUBSND so that the next send is not submitted if the send data set is empty.
   b. Reference the copied or updated BLMSAL1 in step COPY2.

   You only need BLMSASDA on the MVS image where the send database resides because you have deleted the step to submit BLMSASDA from the receive JCL.

   Each time you submit the modified BLMSASDA, the send data set is deleted and one Automatic Log Save Send Function and one Automatic Log Save Receive Function is performed. JCL to start another Automatic Log Save Send Function is not submitted.

Deleting a LOGSAVE Record

If you are propagating data to Information/Management (that is, you are using both the Automatic Log Save Send and Automatic Log Save Receive Functions) and you need to delete the LOGSAVE record on either the send or the receive database, follow the procedures outlined below to preserve the VSAM keys for the LOGSAVE records.

If you are using only the Automatic Log Save Send Function, which is the case if you are propagating data only to DB2, you can delete the LOGSAVE record without regard for preserving the VSAM key.

Deleting a Send LOGSAVE Record

If you must delete the Send LOGSAVE record, decide whether you want to preserve the record’s VSAM key. If you do not preserve the VSAM key and you are propagating data to an Information/Management receive database, you will have to rebuild the receive database.

Preserving the LOGSAVE Record’s VSAM Key

1. Stop send processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.
2. Suspend normal database use.
3. Delete the LOGSAVE record.
4. Recreate the LOGSAVE record.
5. Resume normal database use.
Without Preserving the LOGSAVE Record’s VSAM Key
1. Stop send processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.
2. Stop receive processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.
3. Delete the send LOGSAVE record.
4. Recreate the send LOGSAVE record.
5. Rebuild the receive database. Use one of the procedures in “Copying the LOGSAVE Record into the Receive Database” on page 12-22 to do this.
6. Update the LOGSAVE record on the receive database. See “Updating the LOGSAVE Record for Receive” on page 12-24 for details.
7. Start normal send and receive processing. See “Starting Normal Send and Receive Processing” on page 12-32.

Deleting a Receive LOGSAVE Record
If you must delete the Receive LOGSAVE record, do the following tasks:
1. Stop send processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.
2. Stop receive processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.
3. Delete the receive LOGSAVE record.
4. Rebuild the receive database. Use one of the procedures in “Copying the LOGSAVE Record into the Receive Database” on page 12-22 to do this.
5. Update the LOGSAVE record on the receive database. See “Updating the LOGSAVE Record for Receive” on page 12-24 for details.

Recovering from Errors
This section includes the following procedures for error recovery:
- “Error during Send or Receive Processing” on page 12-36 outlines a general procedure to follow when an error occurs during send or receive processing.
- “Restoring Synchronization” on page 12-37 outlines three procedures for restoring synchronization. You can use one of these procedures, or your site-specific procedure, to restore synchronization.
Error during Send or Receive Processing

If an error occurs during send or receive processing, do the following tasks:

1. Stop send and receive processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.

2. Diagnose the problem:
   a. Examine the output from the BLMSASD, BLMSASDA, or BLMSARV job to determine the nature of the failure.
   b. Examine the SYSPRINT data set on the send database and on the receive database for any messages.
   c. Examine the return and reason codes from the user exit that failed.
   d. Look up the return and reason codes in the table for the user exit:
      - For user exit BLGUT3EX, see Table 36 on page 14-2.
      - For user exit BLGUT3WT, see Table 37 on page 14-3.
      - For user exit BLGUT4EX, see Table 38 on page 14-7.
      - For user exit BLGUT4WT, see Table 39 on page 14-8.

   The description column explains the source of the error and indicates what to do to correct the problem.

3. Correct the problem that caused the error.

4. If a send data set was created (the SDLDS was successfully offloaded) but the copy to the GDG failed:
   a. Manually copy the send data set to the GDG.
   b. Prepare and submit job BLMSASDE to process the send data set. See “Preparing BLMSASDE.” If the receive is successful, normal send and receive processing is resumed.

5. If a send data set was created (the SDLDS was successfully offloaded) and a receive job was submitted but did not process or if the onload was unsuccessful:
   a. Prepare and submit job BLMSASDE to process the send data set. See “Preparing BLMSASDE.” If the receive is successful, normal send and receive processing is resumed.

Preparing BLMSASDE

1. Change the job card to your local standards.

2. Change the node.userid on the /*ROUTE PRINT card to the node and user ID to receive the job output.

3. In step XMITSND, change userid.dsname1 to the name of the send data set or generation of a GDG that you want to process.

4. In step COPY1, change:
a. The JCL data set name to the name of your local data set that contains JCL BLMSARV.
b. The space DCB parameter on the SYSUT2 DD name to a value large enough to handle splitting each 80-byte record created from the transmit into two 80-byte records (the splitting is performed in step SPLIT).

5. In step COPY2, change the JCL data set name to the name of your local data set that contains BLMSAL1.

6. In step GOODSUB, change:
   a. sample message to your message for a successful submission of the receive JCL.
   b. node.userid to the destination for the message.

Restoring Synchronization

Synchronization is the process by which the Automatic Log Save Facility ensures that the send data sets are loaded into the receive database in the same order that they were offloaded from the send database. Synchronization can provide additional protection against loading data out of order.

The Automatic Log Save Facility stores the sequence number of the send data set in the following places to assist you in recovering from errors:

- LAST(SDLDS SENT) key in the SDIDS of the send database
- LAST(SDLDS SENT) key in the SDIDS of the receive database
- LAST(SDLDS SENT) record in the send and receive data sets.

See “Maintaining Synchronization” on page 11-22 for more information about these records and how to determine their values.

When synchronization processing is on (Synchronize Send and Receive? field is YES), the following processing occurs before records are converted for storage in the receive database:

- The sequence number stored in the LAST(SDLDS SENT) record in the SDIDS of the receive database is read and incremented by 1.
- That this sequence number matches the sequence number in the LAST(SDLDS SENT) record in the receive data set is verified.

If the values do not match, send and receive processing are out of synchronization and receive processing stops.

If the Automatic Log Save Receive function stops because send and receive processing are out of synchronization, user exit BLGUT3EX writes a TSCAFRET of 8 and a TSCAFRES of 20 to the data set allocated to the SYSPRINT DDNAME on the receive database. Normally, this does not happen because the next send is only started after a successful receive. You could, however, inadvertently start another send before a previous send data set is received, for example, while recovering from a receive error. To restore synchronization, you can process the generations of your GDG that were either processed incorrectly or not at all. If you do not have GDGs, you can copy your entire send database over to your receive database and
rebuild the SDIDS on the receive database. You can use BLGUT1 to rebuild the SDIDS on the receive database, or you can copy the entire SDIDS from the send database. Using BLGUT1 to rebuild the SDIDS requires copying only the LAST_SDLDS_SENT record from the send database, so your send database is out of production for less time if you use BLGUT1.

You can use one of the following procedures, or a procedure developed and tested at your site, to restore synchronization:

- "Restoring Synchronization with GDG Data Sets." Use this procedure if you are copying send data sets to generations of a GDG.
- "Restoring Synchronization Using BLGUT1" on page 12-41. Use this procedure if you are not copying the send data set to a generation of a GDG, or if the generations with the correct sequence numbers are not available and you want to minimize lost production time on your send database.
- "Restoring Synchronization without Using BLGUT1" on page 12-42. Use this procedure if you are not copying the send data set to a generation of a GDG, or if the generations with the correct sequence numbers are not available and you do not want to use BLGUT1.

All of the procedures require you to use IDCAMS. For information about IDCAMS, refer to the following publications:

- MVS/DFP Access Method Services for the Integrated Catalog Facility
- MVS/DFP Macro Instructions for VSAM Data Sets
- MVS/DFP Using Data Sets

After restoring synchronization and before resuming normal send and receive processing, determine why synchronization was lost.

**Restoring Synchronization with GDG Data Sets**

After you complete this procedure, the receive database contains the records from the send data sets that were created by the Automatic Log Save Send Function but that were not successfully received by the Automatic Log Save Receive Function. Also, the value stored in the LAST_SDLDS_SENT record in the receive database matches the value stored in the LAST_SDLDS_SENT record in the send database. Therefore, synchronization is restored.

To use this procedure, you must be saving copies of the send data set in a GDG and all the send data sets that were not successfully processed by the Automatic Log Save Receive function must be available.

**Note:** The program administrator working on the receive side needs two user IDs available that use the session-parameters member with read/write access to the receive database.

1. On the MVS image where the send database resides, stop processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.

2. On the MVS image where the send database resides, check the SYSPRINT output to verify that user exit BLGUT4EX returned a TSCAFRES of 0 and a
TSCAFRET of 0. (This verifies that the sequence number was incremented and is 1 greater than the number in the send data set. It also verifies that there was not a problem on the send database that you need to consider.)

3. From the Information/Management Primary Options Menu on the send database, issue the IRC glossary, up max, find last to determine the sequence number stored in the LAST_SDLDS_SENT record of the SDIDS.

4. From the Information/Management Primary Options Menu on the receive database, issue the IRC glossary, up max, find last to determine the sequence number of the last receive data set that was received and processed by the receive database. This number is stored in the LAST_SDLDS_SENT record of the SDIDS.

5. To determine how many send data sets were not received, subtract the sequence number stored in the LAST_SDLDS_SENT record of the receive database’s SDIDS, which you determined in step 4, from the sequence number stored in the LAST_SDLDS_SENT record of the send database’s SDIDS, which you determined in step 3. The result is the number of data sets that have not been processed.

Using the example in the first row of Table 35 on page 12-40, where the sequence number from the send database is 12 and the sequence number from the receive database is 7, you determine that five data sets have not been received.

6. To determine which data sets have not been received, use the value for how many data sets have not been received and the sequence number from the receive database. Using the example in the first row of Table 35 on page 12-40, you determine that data sets with sequence numbers 8, 9, 10, 11, and 12 are the five data sets that need to be sent and received to restore synchronization. After the data set with sequence number 12 is processed on the receive database, the sequence number in the receive database will be 12. At that point, the sequence number in the send database will match the sequence number in the receive database, and normal receive and send processing can be resumed. Table 35 on page 12-40 gives additional examples of determining how many data sets were not received and their sequence numbers.
Recovering from Errors

7. On the MVS image where the send database resides, find the GDG data sets that contain the sequence numbers that were not received. For the example in the first row of Table 35, you need to find the data sets that contain the sequence numbers 8, 9, 10, 11, and 12.
   a. Browse the most recently created GDG data set.
   b. Turn on the hex display.
   c. The first record in the data set contains the LAST(SDLDS_SENT) record in hex. See “Maintaining Synchronization” on page 11-22 for details on reading this key.
   d. Convert the hex value to decimal.

8. After you locate the correct GDG data sets:
   a. Copy and modify JCL BLMSAL1 so that it does not submit BLMSASDA after the receive.
   b. Copy and modify JCL BLMSASDE. See “Preparing BLMSASDE” on page 12-36. Supplement those instructions with the following:
      1) In step XMITSND, specify the entire data set name of the lowest sequence number of the GDG, for example, userid.gdg.G0001V00, on the TRANSMIT statement. For the example in the first row of Table 35, this is the data set that contains the sequence number 8.
      2) In step BLDJOB, replace BLMSAL1 with the name you saved JCL BLMSAL1 as in step 8a.

9. On the receive database, change the value in the Time interval in minutes field in the LOGSAVE record to 0, if it is not already 0.

10. Submit the modified version of BLMSASDE to process the specified data set and perform the receive function.

11. Check the job output for errors. If the job output contains no errors, receive processing for the first GDG data set is finished and the sequence number in the receive database was incremented.
12. Modify and submit BLMSASDE again. This time specify the next GDG data set. For the example in the first row of Table 35, this is the data set that contains the sequence number 9.

13. Repeat steps 11 on page 12-40 and 12 until all the GDG data sets needed to restore synchronization have been processed. For the example in the first row of Table 35 on page 12-40, this is 3 more times to process the data sets that contain sequence numbers 10, 11, and 12.

14. To verify that the LAST_SDLDS_SENT records on the receive database and the send database both contain the same value, issue the GLOSSARY command on both the send database and the receive database as you did in steps 3 and 4 on page 12-39. For the example in the first row of Table 35 on page 12-40, both records should now contain 12.

15. On the receive database, change the value in the Time interval in minutes field in the LOGSAVE record to the value for normal receive processing.


**Restoring Synchronization Using BLGUT1**

After you complete this procedure the receive database contains copies of the records that were in the send database at the time the REPRO was run. Also, the value stored in the LAST_SDLDS_SENT record in the receive database matches the value stored in the LAST_SDLDS_SENT record in the send database. Therefore, synchronization is restored.

1. Stop send processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.

2. On the receive database, print the LOGSAVE record. You will use this print out in step 4h on page 12-42 to recreate the LOGSAVE record.

3. On the MVS image where the send database resides, do the following tasks:
   a. Issue the BLX-SP FREE command on the send database’s SDDS and SDIDS so that no records can be created on the send database. Refer to the Operation and Maintenance Reference for information on the BLX-SP commands.
   b. REPRO the send database’s SDDS.
   c. REPRO the LAST_SDLDS_SENT record from the SDIDS. You can use the following IDCAMS control statements to copy the LAST_SDLDS_SENT record.

   ```sql
   REPRO IDS('your.SDIDS') ODS('your.outrec') +
   FROMKEY(X'BAD3C1E2E36DE2C4D3C4E26DE2C5D5E35C') +
   COUNT(1)
   ```
   d. Issue the BLX-SP REALLOC command on the send database’s SDDS and SDIDS.
e. Do not start send processing, but you may start using your send database for other functions.

4. On the MVS image where the receive database resides, do the following tasks:
   a. Delete and redefine the receive database’s SDIDS and SDDS.
      1) Issue the BLX-SP FREE command on the receive database’s SDDS and SDIDS.
      2) Use IDCAMS to delete the receive database’s SDIDS and SDDS.
      3) Use IDCAMS to define the receive database’s SDIDS and SDDS.
   b. REPRO just the SDDS from the send database into the receive database.
   c. Issue the BLX-SP REALLOC command on the receive database’s SDDS and SDIDS.
   d. Run BLGUT1 on the receive database’s SDIDS. Refer to the Operation and Maintenance Reference for more information.
   e. Issue the BLX-SP FREE command on the receive database’s SDIDS.
   f. REPRO the LAST(SDLDS SENT) record from the send database into the receive database’s SDIDS. (This is the record you copied in step 3c on page 12-41.) You have to do this step because BLGUT1 does not create the LAST(SDLDS SENT) record and because you need the value stored in this record on the receive database to match the value stored in this record on the send database.
   g. Issue the BLX-SP REALLOC command on the receive database’s SDIDS.
   h. Update the receive database’s LOGSAVE record. Change the Database type field from send to receive. Refer to the print out of your LOGSAVE record, which you made in step 2 on page 12-41, to determine whether you need to make any other changes.

5. Start normal send and receive processing. See “Starting Normal Send and Receive Processing” on page 12-32.

**Restoring Synchronization without Using BLGUT1**

After you complete this procedure the receive database contains copies of the records that were in the send database at the time the REPRO was run. Also, the value stored in the LAST(SDLDS SENT) record in the receive database matches the value stored in the LAST(SDLDS SENT) record in the send database. Therefore, synchronization is restored.

To restore database synchronization, do the following

1. Stop send processing. See “Stopping Send and Receive Processing” on page 12-32 for details on how to do this.

2. On the receive database, print the LOGSAVE record. You will use this print out in step 4d on page 12-43 to recreate the LOGSAVE record.

3. On the MVS image where the send database resides, do the following tasks:
Propagating to a Database

Recovering from Errors

a. Issue the BLX-SP FREE command on the send database’s SDDS and the SDIDS so that no records can be created on the send database. Refer to the *Operation and Maintenance Reference* for information on the BLX-SP commands.

b. REPRO the send database’s SDDS and SDIDS.

c. Issue the BLX-SP REALLOC command on the send database’s SDDS and SDIDS.

d. Do not start send processing, but you may start using your send database for other functions.

4. On the MVS image where the receive database resides, do the following tasks:

   a. Delete and redefine the receive database’s SDIDS and SDDS.
      1) Issue the BLX-SP FREE command on the receive database’s SDDS and SDIDS.
      2) Use IDCAMS to delete the receive database’s SDIDS and SDDS.
      3) Use IDCAMS to define the receive database’s SDIDS and SDDS.

   b. REPRO the SDDS and SDIDS from the send database into the receive database.

   c. Issue the BLX-SP REALLOC command on the receive database’s SDDS and SDIDS.

   d. Update the receive database’s LOGSAVE record. Change the Database type field from send to receive. Refer to the print out of your LOGSAVE record, which you made in step 2 on page 12-42, to determine whether you need to make any other changes.

5. Start normal send and receive processing. See “Starting Normal Send and Receive Processing” on page 12-32.
Propagating to a DB2 Database

This chapter contains Product-sensitive Programming Interface and Associated Guidance Information.

This information is for the program administrator responsible for implementing the DB2 Extract Facility. You must understand Terminal Simulator Panels (TSPs), the Panel Modification Facility (PMF), and BLGUT8 to perform the tasks described in this chapter. For information about TSPs, refer to the Terminal Simulator Guide and Reference. For information about PMF, refer to the Panel Modification Facility Guide. For information about BLGUT8, refer to the Operation and Maintenance Reference.

Some of the tasks described in this chapter also require a knowledge of DB2. Work with your DB2 database administrator to accomplish those tasks. You may need to refer to the following books:

- IBM Database 2 Version 2 Administration Guide
- IBM Database 2 Version 2 SQL Reference
- IBM Database 2 Version 2 Messages and Codes
- IBM Database 2 Version 2 Command and Utility Reference

Implementing the DB2 Extract Facility

Before you implement the DB2 Extract Facility, you must understand the process and perform the following tasks:

1. Read “Propagating Information/Management Data” on page 11-1 for a discussion of data propagation and an overview of the LOGSAVE record.

2. Read this chapter for an overview of DB2 Extract Processing and the implementation tasks.

3. Decide how your installation will use the DB2 Extract Facility:
   - How often do you want to offload the SDLDS of the send database?
   - Do you want to change how long an enqueue on the SDLDS is held?
   - Which Information/Management records and fields will you propagate to DB2?
   - Which user IDs will receive status messages indicating whether the update step of the DB2 Update Utility processed successfully?
■ Which user ID will the job output from the DB2 Update Utility be routed to?

■ How often will you review the status messages from DB2 and how will you handle error situations?

■ Will you maintain copies of the offloaded SDLDS (send data set) in a generation data group (GDG)?

■ Will you maintain copies of the SQL data set in a GDG?

■ If you maintain GDGs, how many backup levels will you keep to ensure adequate data integrity? Consider the frequency of send processing in setting this value.

■ Do you want to synchronize the DB2 Extract Send and Update Utilities?

■ Do you want to disable automatically starting the next DB2 Extract Send operation when the DB2 Update completes successfully?

■ What procedures will you use for backup and disaster recovery?

■ How will the automatic offload of the SDLDS fit in with your current database recovery procedures?

■ Do you want to propagate Information/Management data to both DB2 and Information/Management? If so, how will your Automatic Log Save processing fit in with your DB2 Extract processing?

4. Develop and test your backup and disaster recovery procedures.

5. Develop procedures for monitoring the operation of the DB2 Extract Facility.

**Understanding DB2 Extract Send Processing**

The DB2 Extract Send Utility reads the LOGSAVE record to obtain the following information:

■ Sequential data set characteristics for the Automatic Log Save send data set and for the SQL data set, including:
  - Data set name
  - Logical record length
  - Block size
  - Primary allocation blocks
  - Secondary allocation blocks
  - Unit type
  - ABEND disposition

■ Database type

■ How long to wait between send processing operations

■ Whether to tag the offload data set with a sequence number so that the DB2 Extract Send and Update Utilities can be synchronized

■ Authorization ID of the DB2 tables
Propagating to a DB2 Database

Understanding DB2 Extract Send Processing

- Names of the RDMTs that map the Information/Management records to DB2
- Whether send processing is enabled or disabled.

The DB2 Extract Send Utility is started by JCL BLMSDSD, which does the following tasks:

1. Starts Information/Management in batch mode and starts TSP BLGTDBXM, which does the following tasks. See “Data Propagation User Exits” on page 14-1 for more information about the user exits.
   a. Calls user exit BLMSSGEN to setup the environment and read the RDMTs that were created by BLGUT8 and stored in a data set in the BLGRFT concatenation.
   b. Calls user exit BLGUT4WT to:
      1) Read the LOGSAVE record.
      2) Verify that the LOGSAVE record is for a send database.
      3) Verify that the status of the LOGSAVE record is enabled.
      4) Put selected information from the LOGSAVE record into the TSCA variable data area.
      5) Wait the length of time specified in the LOGSAVE record; then return to the TSP.
   c. Call user exit BLGUT4EX to:
      1) Read information from the TSCA variable data area.
      2) Allocate the send data set.
      3) If you are synchronizing the send and receive functions:
         a) Obtain the sequence number from the SDIDS.
         b) Add 1 to the sequence number.
         c) Write the sequence number as the first record in the send data set.
      4) Move records from the SDLDS into the send data set and empty the SDLDS.
      5) If you are synchronizing the Send and Update Utilities, store the sequence number in the SDIDS.
   d. Calls user exit BLMSSGEN to do the following until all the records in the send data set have been read:
      1) Read a record from the send data set.
      2) Match a record type s-word from an RDMT to the record. Use the first RDMT for which the record type s-word is found in the Information/Management record.
      3) If no RDMT is found that maps the record, go to the next record.
      4) If a match, build SQL statements for the record using the RDMT to determine which fields to extract and the information that is needed to construct the SQL statement.
      5) Write the SQL statements to the SQL data set.
Understanding DB2 Extract Send Processing

e. If no SQL was built and no errors occurred, submits JCL BLMSDSD to begin another send.

f. Submits JCL BLMSSND, which builds and submits jobstream BLMSRCV by concatenating the following pieces:
   1) JCL member BLMSRCV
   2) The SQL statements from the SQL data set as in stream data
   3) JCL member BLMSL1.

   The resultant job (BLMSRCV) is submitted to the internal reader and routed to the MVS image specified on the /*ROUTE XEQ card, where it runs the SQL in batch mode to update the DB2 database. See “Understanding DB2 Update Processing” for a discussion of the processing done by BLMSRCV.

g. Calls user exit BLMSSGEN to cleanup the environment.

h. Submits JCL BLMSDCS to copy the send data set to a GDG based upon the success or failure of step 1c on page 13-3.

i. Send is complete.

2. Sends a status message about the success or failure of the TSP to the user ID specified in the JCL.

Understanding DB2 Update Processing

The DB2 Update Utility uses BLMSRCV, the JCL generated and submitted by the DB2 Extract Send Utility. This JCL does the following tasks:

1. Starts DSNTEP2, a PL/I program provided by DB2. This program:
   - Runs the in-stream SQL statements in batch mode to update the data in the DB2 database.
   - If you leave SYSPRINT DD SYSOUT=* in the JCL, messages are written to SYSPRINT on the MVS image where DB2 resides. If you specified a data set for this DD name, writes messages to that data set on the MVS image where DB2 resides. Refer to the prologue of DSNTEP2 for a list of the messages it returns.

2. Sends a status message to the user ID specified in the JCL.

3. Submits JCL BLMSDSD to begin the next send. This job is routed to the MVS image where the send database resides for execution.

4. Routes the job output to the user ID specified on the /*ROUTE PRINT card in the JCL.
Setting Up the DB2 Extract Facility

To propagate Information/Management data only to a DB2 database, do the following tasks. You can do these tasks in any order you choose. You must do all of them, however, before you start TSP BLGTDBXM.

1. Be sure the Information/Management database is running with an SDLDS.

2. Build relational data mapping tables (RDMTs) to map Information/Management records to DB2. See “Building Relational Data Mapping Tables” for information on this task.

3. Define the LOGSAVE record for DB2 Extract processing. See “Defining a LOGSAVE Record for DB2 Extract Processing” on page 13-13 for information on this task.

4. Establish the DB2 environment. See “Establishing the DB2 Environment” on page 13-18 for information on this task.

5. On the MVS image where DB2 resides, locate and modify your local copy of DSNTEP2. See “Modifying DSNTEP2” on page 13-21 for information on this task.

6. Create GDGs. See “Creating GDGs” on page 13-22 for information on this task.

7. Prepare the JCL that is used for DB2 Extract processing. See “Preparing the JCL” on page 13-23 for information on this task.

8. Modify the TSP (BLGTDBXM) that performs DB2 Extract processing. See “Modifying the DB2 Extract TSP” on page 13-29 for information on this task.

9. Submit JCL BLMSDSD to initiate data propagation to DB2. See “Starting Send and Update Processing” on page 13-31 for more information on this task.

If you are propagating data to both Information/Management and DB2, you must make some additional changes to TSP BLGTDBXM and to the JCL for the Automatic Log Save Facility and the DB2 Extract Facility. Those changes are described in “Propagating Data to Both Information/Management and DB2” on page 13-35.

Building Relational Data Mapping Tables

Use BLGUT8 to build tables that map Information/Management panels and fields to relational database tables and columns. These tables are called relational data mapping tables (RDMTs). The DB2 Extract Send Utility uses RDMTs to construct the SQL statements for propagating Information/Management data to DB2.

BLGUT8 is used to build tables for both the DB2 Extract Facility and the Information/Management application program interfaces (APIs). If you are building an RDMT, any keywords that apply only to low-level application program interface (LLAPI) program interface data tables (PIDTs) are ignored. If you are building a PIDT, any DB2 Extract Facility related keywords are ignored. Therefore, if you are using both the DB2 Extract Facility and one of the APIs, you can use the same input statements to build both PIDTs and RDMTs. Refer to the Application Program
Interface Guide for details on using BLGUT8 and the syntax for creating PIDTs. The BLGUT8 syntax described here applies only to building RDMTs.

If you add or delete fields, or change their lengths, in the Information/Management record after you create an RDMT for it, consider whether you need to rerun BLGUT8 to update the Information/Management to DB2 map. If the map changes, you need to consider altering, or deleting and re-creating, the DB2 tables to conform to the new format for the Information/Management record.

BLGUT8 Syntax

The TABLE statement accepts SQLMAP as a value for USE. Specifying USE(SQLMAP) tells BLGUT8 to generate an RDMT for each TABLE and ETABLE pair you specify as input.

The FIELD statement accepts the keywords SQLTAB and SQLCOL. These keywords are ignored if USE(SQLMAP) is not specified. If USE(SQLMAP) is specified, keywords that are not used, such as NOTLOGIC and REQUIRED, are ignored. This enables you to use the same statements to create either PIDTs for the API or RDMTs for the DB2 Extract Facility. Refer to the Application Program Interface Guide for information on creating PIDTs.

To create an RDMT, specify USE(SQLMAP) on the TABLE statement. Specify the name of the DB2 table with the SQLTAB keyword on the FIELD statement. Specify the DB2 column with the SQLCOL keyword on the FIELD statement.

TABLE/ETABLE Statement: You can specify multiple RDMTs in one input stream by using the TABLE and ETABLE combination. SQLMAP statements make up a record and specify the mapping of the record to tables and columns in a relational database.

Keywords on the TABLE statement specify table attributes, and the TABLE and ETABLE combination defines the boundaries of a table specification. Each table specification corresponds to a record or search argument definition and generates one data table and one validation table. All other statements must lie within a TABLE and ETABLE combination.

The syntax of the TABLE statement is:

\[
\begin{align*}
[\text{label:}] & \quad \text{TAble Name(name) Use } \{(\text{Inquiry}) \} \\
& \quad \{(\text{Retrieve}) \} \\
& \quad \{(\text{Create}) \} \\
& \quad \{(\text{Update}) \} \\
& \quad \{(\text{Add}) \} \\
& \quad \{(\text{Alias}) \} \\
& \quad \{(\text{Header}) \} \\
& \quad \{(\text{Sqlmap}) \} \\
\end{align*}
\]

\[
\begin{align*}
[\text{label:}] & \quad \text{ETable} \\
\end{align*}
\]

Figure 16. TABLE Statement Syntax
Name
Identifies the name to be given to the table. This name also serves as the PDS member name. You can use up to 8 characters for the table name when the value specified with Use is ALias or Sqlmap. You can use up to 7 characters for all other tables. Two member names are determined by this specification. The PIDT is named with the value for this keyword, and the PIPT name is this value with a P appended. This keyword determines PIDTNAME, PIDTPTNM, and PIPTNAME.

Use
Identifies the function for which the table is used. The value specified with this keyword determines how the APIs set PIDTUSEF for all USE keyword values except ALias and Sqlmap. Use(ALias) causes the Table Build Utility to process an alias table only. Use(Sqlmap) causes the Table Build Utility to process a relational data mapping table only. Use(Header) causes the Table Build Utility to process header rows only.

Inquiry
For fields to include in a search argument.

Retrieve
For fields to retrieve from an existing record.

Create
For fields to include in a new record.

Update
For fields to update in an existing record.

Add
For record relation fields to add to an existing record.

ALias
For fields to collect for an alias table. You can use only ALias statements when you specify Use(ALias).

Header
For fields to create a model PIDT containing only header rows. This PIDT is for use in generating dynamic PIDTs on the retrieve transaction (T100). A FIELD statement specified with Use(Header) is ignored.

Sqlmap
For fields to map to relational database tables and columns.

FIELD Statement: The FIELD statement defines the Information/Management fields that are to be extracted from the log data set (the SDLDS) and mapped to DB2 tables and columns. A field is any attribute that can be assigned to a record type and is characterized by an s-word or a p-word, or both. Examples are reporter name, record ID, freeform text description, and date entered.

For a Use (Sqlmap table, do not include a FIELD statement for the RNID. The DB2 Send Utility automatically generates SQL statements for this field.

The syntax of the FIELD statement as used by the DB2 Extract Utility is:
[label:] Field Panel(panel) Index(dictionary index key)

[SQLTAB(table-name)] [SQLCOL(column-name)]

[RCdsword {(Yes)}] [List {(Yes)}] [Text {(Yes)}]

{(No)} {(No)}

Figure 17. FIELD Statement Syntax

If you omit a keyword from the FIELD statement, the default for the keyword is NO. However, if you include the keyword in your statement, you must also enter YES or NO.

Panel
Identifies the panel in which the Information/Management field is located. This keyword is required.

panel
An 8-character alphanumeric string. The first character must be alphabetic (A-Z). This string names a member of a panel data set.

Index
Identifies the Information/Management field. This keyword is required.

dictionary index key
The character P or S followed by 4 hexadecimal characters, which identify a p-word or an s-word, respectively, in the dictionary data set or specified panel. This s-word index or p-word index must be in the panel named by the PANEL keyword unless you specify an assisted-entry panel with its Collect From Caller field set to YES. In this case, the s-word is obtained from the dictionary.

For control panels, the DB2 Extract Facility uses only ADD control lines. Within these control lines, it uses only p-words with literal p-word validation data (pfx/<data> or <data>).

When you specify an s-word index key and control panel, the DB2 Extract Facility retrieves corresponding literal prefix validation data, if available, along with the s-word. If you specify an s-word index key and an assisted-entry panel, the DB2 Extract Facility uses the first p-word that appears in the validation section of the specified assisted-entry panel to define the field.

For a Use (Sqlmap table, BLGUT8 suppresses any entry that it determines does not have actual data. Two examples of this are:

1. An s-word that defines a visible phrase from a selection or data-entry panel
2. An s-word that is added by a control panel with no corresponding data.

Other cases may cause this situation. These entries are not listed in the report and no SQL is generated by the DB2 Extract Send Utility for them. All entries put into the RDMT will have SQL generated for them. If a MAXL of 0 appears in the output, BLGUT8 did not have enough information to
determine what the maximum length for the field is. You must manually determine what the maximum length of the field is by looking at the panel that collects the data.

The DB2 Extract Send Utility tries to find the data for an RDMT entry in the Information/Management record. If it does not find any data for the entry, it uses the MAXL field to determine how many blanks to use as the data. If MAXL is 0, it supplies 1 blank as the data for the field.

You can only specify a p-word index key (Pxxxx) for an assisted-entry or control panel. Specify a p-word key only if you are describing a field that does not have an s-word.

Specify each index keyword value only once within a TABLE and ETABLE section.

**SQLTAB**
Identifies the DB2 table in which the Information/Management field is to be placed. If this keyword is omitted and the LIST(Y) or TEXT(Y) is not specified, the DB2 table to which the field is mapped defaults to the value of NAME given on the TABLE statement. If this keyword is omitted and the LIST(Y) or TEXT(Y) is specified, the table name defaults to the value of NAME given on the TABLE statement concatenated with an underscore character followed by the dictionary index key.

`table-name`
This string names the DB2 table in which this Information/Management field is to be placed. This can be from 1 to 18 characters long. The data that you enter in this field is not validated. Ensure that this field contains a valid DB2 table name. If the name is not valid, the DB2 Update Utility processes with errors.

**SQLCOL**
Identifies the DB2 column in which the Information/Management field is to be placed. If this keyword is omitted, the column to which the field is mapped defaults to the dictionary index key.

`column-name`
This string names the DB2 column in which this Information/Management field is to be placed. This can be from 1 to 18 characters long. The data that you enter in this field is not validated. Ensure that this field contains a valid DB2 column name. If the name is not valid, the DB2 Update Utility processes with errors. You must map each list and text entry to a unique table.

**List**
Identifies list processing for the field.

**Yes**
Indicates a list field. Specify List(Y) only when the Panel keyword value specifies an assisted-entry panel and the Index keyword value is an s-word.
index (Sxxxx) that refers to an s-word of no more than 8 characters.
List(Y) is not allowed with Text(Y) or RCdsword(Y).

**No (default)**
Indicates a nonlist field.
If you omit this keyword, the default is NO. If you include the keyword,
specify either YES or NO.

**Text**
Identifies a freeform text field.

**Yes** Indicates a freeform text field. Specify Text(Y) only when the Panel
keyword value names a selection, option, control, or data-entry panel.
Text(Y) is not valid when the Index keyword value represents a p-word
index (Pxxxx). Do not use Text(Y) with List(Y) or RCdsword(Y).

**No (default)** Indicates a field that does not identify freeform text.
If you omit this keyword, the default is NO. If you include the keyword,
specify either YES or NO.

**RCdsword**
Identifies the field that defines the record type.

**Yes** Indicates a record type field. Specify RCdsword(Y) only when the Panel
keyword value names a selection, option, data-entry, or control panel.
RCdsword(Y) is not valid when the Index keyword value represents a
p-word index (Pxxxx).

You must specify RCdsword(Y) for one and only one Sqlmap statement
within a table specification. RCdsword(Y) is not allowed with List(Y) or
Text(Y).

**No (default)** Indicates a field that does not define the record type.
If you omit this keyword, the default is NO. If you include the keyword,
specify either YES or NO.

**RDMT Build Job Stream Example**
Figure 18 on page 13-11 shows how to use BLGUT8 to build an RDMT named
MYMAP. In this example, the RDMT is stored in a partitioned data set named
XXX.INFOV6.SBLMFMT.PDS referenced by the DDname BLGRFT. The actual DDname
to use is the one your session-parameters member uses for the RFT data set. Only a
portion of the problem record type fields are included.

In the job stream, DDname BLGPNLS references the panel data set that was created
by using BLGUT6 to copy the panels into a partitioned data set. DDname BLGDICT
references the dictionary data set.

The sample library SBLMSAMP contains input statements that build PIDTs. You can
modify these statements to build RDMTs.
PROPAGATING TO A DB2 DATABASE

SETTING UP THE DB2 EXTRACT FACILITY

COUNT JOB EXEC PGM=BLGUT8,REGION=248K
//STEP1 EXEC PGM=BLGUT8,REGION=248K
//STPLIB DD DSN=xxx INFOV6 SBLMMOD1 DISP=SHR
//BLGPNLS DD DSN=xxx INFOV6 SBLMPNLS DISP=SHR
//BLGDICT DD DSN=xxx INFOV6 DICT DISP=SHR /* VSAM */
//BLGRFT DD DSN=xxx INFOV6 SBLMFMT.PDS DISP=OLD
//SYSPRINT DD SYSOUT=A

;~CHR='DLM=%%' CHANGES END-OF-STREAM DELIMITER FROM '/*;' TO '%%'
;~CHR OTHERWISE, COMMENT STARTED IN FIRST COLUMN TERMINATES INPUT STREAM
//SYSIN DD *,DLM='%%'

TABLE NAME(MYMAP) USE(SQLMAP);

FIELD PANEL(BLG00000) INDEX(S0032) /* PROBLEM RECORD TYPE */
RCDWORD(Y);

FIELD PANEL(BLG6REQN) INDEX(S0859) /* REPORTER NAME */
SQLTAB(PROBLEM_DATA) SQLCOL(REPORTER_NAME);

FIELD PANEL(BLG6PTYP) INDEX(S0C09) /* PROBLEM TYPE */
SQLTAB(PROBLEM_DATA) SQLCOL(PROBLEM>Type);

FIELD PANEL(BLG6RQDP) INDEX(S0B9B) /* REPORTER DEPARTMENT */
SQLTAB(PROBLEM_DATA) SQLCOL(REPORTER_DEPT);

FIELD PANEL(BLG6STAT) INDEX(SO8EE) /* REPORTER STATUS */
SQLTAB(PROBLEM_DATA) SQLCOL(REPORTER_STAT);

FIELD PANEL(BLG6PHON) INDEX(S0B2D) /* REPORTER PHONE */
SQLTAB(PROBLEM_DATA) SQLCOL(REPORTER_PHONE);

FIELD PANEL(BLG6PRII) INDEX(S0B6E) /* INITIAL PRIORITY */
SQLTAB(PROBLEM_DATA) SQLCOL(INITIAL_PRIORITY);

FIELD PANEL(BLG6DSAB) INDEX(S0E0F) /* DESCRIPTION */
SQLTAB(PROBLEM_DATA) SQLCOL(DESCRIPTION);

FIELD PANEL(BLG0B010) INDEX(S0E01) /* TEXT, DESCRIPTION */
SQLTAB(PROBLEM_TEXT) SQLCOL(TEXT) TEXT(Y);

ETABLE; /* END TABLE */

%%

Figure 18. Build RDMT Job Stream

RDMT Job Stream Sample Output

Figure 19 on page 13-12 shows the output that results from a successful run of BLGUT8. Notice that for this example there is no output entry for the FIELD statement defining the record type s-word. This is because the field is a visible phrase type field and does not have any data; therefore, the field is not mapped to DB2. All entries in the output report will have SQL generated for them. In addition, SQL will be generated for the RNID. See “Example of Propagating Problem Data to DB2” on page 13-32 for examples of the DB2 tables this RDMT maps to.
Figure 19. RDMT Output Example

Using the RDMT Output

Use the RDMT output when you create the DB2 tables. Use:

- The table names listed in the SQLTAB column to name the DB2 tables when you create them.
- The column names listed in the SQLCOL column to name the columns in the DB2 tables when you create them.
- The lengths listed in the MAXL column to determine the lengths for the DB2 columns.
- Use the data types listed in the column labeled X to determine the data types for the DB2 columns. Possible data types are as follows:
  - D Double-byte character set (DBCS)
  - M Mixed data
  - S Single byte character set (SBCS)

Use this information to determine the data type for the DB2 column to which this field is mapped. If this column contains a D, map the field to a DB2 column with a data type for double-byte characters. If this column contains an M, consider whether to map the field to a DB2 column with a data type for double-byte characters or single-byte characters. (The M does not mean that the field contains mixed data, only that it can.) Refer to the *IBM Database 2 Version 2 SQL Reference* for information on DB2 data types.

The DB2 table and column names from this report must match the table and column names in the DB2 database. A mismatch will cause the DB2 Update Utility to fail.

“Create Tables to Receive Information/Management Data” on page 13-19 contains more information about how to use this report.
DBCS and Mixed Data Considerations

Information/Management and DB2 both support DBCS data. DSNTEP2, as it is supplied by DB2, does not. If your Information/Management data contains DBCS data, you must modify DSNTEP2 so that it can handle DBCS data.

Defining a LOGSAVE Record for DB2 Extract Processing

This scenario assumes that you have a LOGSAVE record already defined for the Automatic Log Save Send function (the Database type field contains send). If you do not, start at “Creating a LOGSAVE Record for a Send Database” on page 12-7 and create one; then return here.

Note: You cannot propagate Information/Management data from a receive database; you can only propagate Information/Management data from a send database.

To define the LOGSAVE record for DB2 Extract processing, start at the Primary Options Menu and type the following IRC:

```
update r logsave
```

and press Enter.
On BLG0ZU50, the Automatic Log Save Record Summary panel, select **DB2 Extract information**. and press Enter.

Type:

3

```
<table>
<thead>
<tr>
<th>BLG0ZU50</th>
<th>AUTOMATIC LOG SAVE RECORD SUMMARY</th>
<th>RNID: LOGSAVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data set name.............. SEND.LOG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Database type............... SEND Date entered....... 10/19/93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time interval in minutes..... 0060 Time entered....... 15:49</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronize Send and Receive?. YES Date last altered... 10/19/93</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive filter prefix........ ______</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Receive filter data:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SQL data set name........... ____________________________________________</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Status....................... ENABLED</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Select one of the choices, or type END to save or CANCEL to discard changes.
  1. Automatic Log Save data set information.
  2. Automatic Log Save synchronize and filter information.
  3. DB2 Extract information.
  4. Toggle record status (ENABLED/DISABLED).

==> 3
```
The DB2 Extract Information Entry panel contains several required fields, some of which are primed. You can accept these values or change them. You must complete the **Data set name** field, the **Authorization ID** field, and select **Relational Data Mapping Tables**. For this example, name the data set DB2X.SQL and specify `infoapp` as the authorization ID. Type this IRC:

1,db2x.sql,7,infoapp,8,,

and press Enter.
The Relational Data Mapping Table Entry panel appears. Starting at the top and leaving no blanks, list the names of all the RDMTs you want the DB2 Extract Facility to use. (The DB2 Extract Facility stops reading the entries when it detects a blank field.) The RDMTs you list must be in a data set in your BLGRFT concatenation. When you list RDMTs for parent and child records, list child records before their parent records, because a child record typically contains the same s-word as its parent record. If the parent record is listed before the child record, DB2 Extract would find the s-word in the RDMT for the parent before it found the RDMT for the child record and stop looking. Be sure the names you type here match the names you gave to the RDMTs on the BLGUT8 TABLE statements.

After you list all the RDMTs you want to use, type:

end,end,

and press Enter.
The Automatic Log Save Record Summary panel appears. You have finished updating the LOGSAVE record. To file the record, type:

9

and press Enter.

DB2 Extract processing does not actually occur, however, until you start the BLGTDDBXM TSP.
BLG0EN10, the Primary Options Menu, appears with a message that the LOGSAVE record was stored successfully.

For future reference, you may want to print the LOGSAVE record.

**Establishing the DB2 Environment**

The DB2 Extract Facility depends on the existence of the tables you specified when you built the RDMTs. Work with your DB2 database administrator to establish the DB2 environment.

DB2 tables exist in a tablespace; tablespaces, in turn, exist in a database. The first step, therefore, in creating a DB2 table, is to create a DB2 database; the second step is to create a tablespace; and the third step is to create a table and any associated indexes. You can modify the following SQL statement to create a database:

```
CREATE DATABASE database_name
STOGROUP storage_group
BUFFERPOOL buffer_pool
```

You can modify the following SQL statement to create a tablespace:

```
CREATE TABLESPACE tablespace_name
IN database_name
BUFFERPOOL buffer_pool;
LOCKSIZE PAGE
CLOSE YES
DSETPASS data_set_password
PCTFREE 10;
```
Create the Synchronization Table
To maintain synchronization, the DB2 Extract Utility maintains a sequence number in a DB2 table. If the DB2 Extract Facility is to run with synchronization (the LOGSAVE record has YES in the Synchronize Send and Receive? field), you must create this table before you start the facility. See “Maintaining Synchronization” on page 11-22 for a complete discussion of how synchronization is maintained.

You can modify and submit the following SQL statement to create the synchronization table:

```sql
CREATE TABLE authorization_id.SYNCH_TABLE
  (SYNCH_NUMBER INT)
  IN database_name.tablespace_name;
```

If you have been running the Automatic Log Save Facility with synchronization, look in the glossary on the send database for the LAST_SDLDS_SENT key to determine the correct sequence number to enter in SYNCH_TABLE. (See “Maintaining Synchronization” on page 11-22 for more information.) If you have not been using the Automatic Log Save Facility with synchronization, initialize the sequence number to 0. You can use, or modify, this SQL statement to set the sequence number:

```sql
INSERT INTO authorization_id.SYNCH_TABLE(SYNCH_NUMBER)
VALUES(00000000);
```

You can use, or modify, this SQL statement to verify that the sequence number is set to the correct value:

```sql
SELECT * FROM authorization_id.SYNCH_TABLE;
```

**Note:** SYNCH_TABLE must never contain more than one row. If it has more than one row, DSNTEP2 will fail. Therefore, after you initialize SYNCH_TABLE, you must not INSERT another value into this table. (The DB2 Update Utility uses the UPDATE statement to increment the value.)

Create Tables to Receive Information/Management Data
You can use the output report from BLGUT8 to create the DB2 tables to receive the data from each Information/Management record type you mapped. The table must contain a column for each Information/Management field you mapped to it, as listed in the output report. It must also contain a column named RNID, which is not listed in the report.

If you are mapping freeform text, list, or nonreplaceable data, you must create a separate table for each s-word. This table must contain three columns: RNID, SEQ_NUM, and a third column to receive the text, list processor, or nonreplaceable data.

Use the following information about how Information/Management data is mapped to DB2 to construct your SQL CREATE TABLE statements:
The RNID of the Information/Management record is expected to be the primary key to all DB2 tables. (You do not have to define the RNID column as the primary key to your DB2 table; however, this column must be unique.) Do not specify the RNID when you create an RDMT with BLGUT8. The user exit BLMSSGEN assumes that RNID is a column in each table and automatically includes it with each SQL statement. Therefore, your DB2 table must contain a column named RNID even though this column is not listed on the RDMT output from BLGUT8. The RNID column must have a length of 8. The RNID can contain mixed character data; therefore, you must decide whether to define this column for SBCS or DBCS data. Consider the information in “DBCS and Mixed Data Considerations” on page 13-13 when making this decision.

The sequence number is used as a secondary key for list, freeform text, and nonreplaceable data. A column named SEQ_NUM is mapped to the sequence number. Do not specify SEQ_NUM when you create an RDMT with BLGUT8. The user exit BLMSSGEN assumes that SEQ_NUM is a character column in each table that is mapped to a record with list, freeform text, and nonreplaceable data and automatically includes it with each SQL statement it creates for list, freeform text, and nonreplaceable data. Therefore, your DB2 table must also contain a column named SEQ_NUM with a length of 8 and a data type of CHAR even though this column is not listed on the RDMT output from BLGUT8. (Mixed data and DBCS data are not possible for this column.) The table must have a compound key of RNID/SEQ_NUM.

The sequence number referred to here is not a VSAM sequence number, but a sequence number generated by Information/Management for repeatable constructs, such as an instance of freeform text or the row number of an instance of list processor data.

The column length for the column mapped to freeform text must be 250 or the maximum length defined by your Information/Management installation. You must specify a maximum length for the column when you create the DB2 table. See “Example of Propagating Problem Data to DB2” on page 13-32 for an example.

If a specific instance of freeform text, list, or nonreplaceable data is less than the column length and the column is defined as data type CHAR, DB2 pads it with blanks.

If an Information/Management field is defined in an RDMT, but no data is found in the record for that field, the SQL statement generated for that field is blanks with a data length of that defined in the MAXL field of the RDMT. If MAXL is 0, data of 1 blank is used.

For lists, the DB2 Extract Facility stores data and row occurrences, not every blank field.

The DB2 Extract Facility is not sensitive to DB2 indexing. You are responsible for ensuring that, if a column is designated to be unique in DB2 by a unique index, the data coming from Information/Management is unique. The DB2...
Extract Facility does not check for a value’s existence in a column before putting a record into DB2.

- Create all the DB2 tables with the same authorization ID you specified on the LOGSAVE record, for example INFOAPP.

- You can specify the p-word or the s-word index associated with the Information/Management data item to be mapped to DB2. If you specify a p-word and the panel for the field has an s-word, the s-word is used to identify the data. If the field is replaceable Information/Management data, the first occurrence of the data is extracted from the Information/Management record.

- You cannot map the same s-word/p-word combination to multiple DB2 tables. This means that the same s-word/p-word combination cannot be used as a foreign key in multiple DB2 tables.

- You cannot mix different record types in the same DB2 table. For example, you cannot map some data from the change record and some data from a problem record to the same DB2 table.

- SRC records are not supported. User exit BLMSSGEN skips them.

- For purged records, DELETE statements are generated for each DB2 table identified in the RDMT.

- Each Information/Management record is mapped to one or more DB2 tables. A DELETE statement and an INSERT statement are created for each DB2 table to which data is propagated. So the existing row of data is deleted, then the new one is inserted.

- A COMMIT statement is generated after the statements for each RNID. If the record maps to more than one DB2 table, the COMMIT occurs after all SQL statements for the RNID are complete.

- You can put an Information/Management time field in a DB2 column with a TIME data type.

- Dates in the normal Information/Management format of yy/mm/dd do not go into DB2 if you define the DB2 column with a DATE data type. If you want to use the DATE data type for your DB2 column, you need to use an exit routine that converts the date coming from Information/Management into a valid format for DB2 dates. Refer to the IBM Database 2 Version 2 Administration Guide for information on writing and using a date/time exit routine.

**Modifying DSNTEP2**

On the MVS image where DB2 resides, modify DSNTEP2. DSNTEP2 is a DB2 sample application that is provided with the DB2 product. As DSNTEP2 is shipped, it stops after 10 errors. To enable synchronization, you must modify DSNTEP2 so that it stops processing after only 1 error. Even if you do not synchronize send and receive processing, you may want receive processing to stop after only 1 error. This program may require additional modifications to accommodate your data propagation needs. Refer to the IBM Database 2 Version 2 Administration Guide.
for more information on this program. Consult with your DB2 database administrator before attempting any modifications.

To modify DSNTEP2, do the following steps:

1. Locate your locally installed version of DSNTEP2, which was shipped with the current version of DB2. Modify it as follows:
   a. Copy the code to a source library under your control.
   b. Edit the code and change the INIT value on the DECLARE statement for MAXERROR from 10 to 0.
   c. Edit the comment in the PROGRAM SIZES section of the prologue stating the new MAXERROR value.
   d. Make any other changes to the customizable variables listed in the PROGRAM SIZES section of the prologue according to your local installation preferences.

2. Modify the JCL member BLMSTEP2 in the sample library SBLMSAMP according to the prologue. (Observe the directions in “Directions for Updating JCL” on page 13-25.) “Preparing the JCL” on page 13-23 and “Preparing BLMSTEP2” on page 13-28 contain more information about JCL member BLMSTEP2.

3. Submit the modified JCL to compile, link, and bind a new plan for DSNTEP2.

4. The load module you just created is the one you must specify in the JOBLIB statement of JCL member BLMSRCV for userid.loadlib.

   Note: Plans beginning with DSN are usually generated by DB2; so it is recommended that you use a different plan name, perhaps LDSNTEP2 for Local DSNTEP2. Specify this name for plan_name in JCL members BLMSL1 and BLMSL2.

Creating GDGs

This section pertains to creating the generation data group (GDG) bases for the Automatic Log Save send data set and the SQL data set. Jobs BLMSSND and BLMSDCS do the actual backups to the GDGs you create in this procedure.

When you create the GDG bases, you define how many generations of the GDG to keep by the value you specify on the LIMIT parameter.

To create a GDG base for the Automatic Log Save send data set, modify and submit JCL member BLGDG1 from the sample library, SBLMSAMP. Follow the steps outlined in “Directions for Updating JCL” on page 13-25 when updating this member in addition to the following steps:

1. Change the job card to your local standards.

2. On the BLGDG card, change userid.dsname.gdg to the name you want to use for the generation data group for the send data set. Change SPACE and DCB attributes to match those of the send data set.
3. On the DEFINE statement, change `userid.dsname.gdg` to the name you want to use for the generation data group for the send data set. This name must match the name you used on the BLGDG card.

4. Change any other options in the DEFINE statement at your discretion. For example, you may want to increase or decrease the value for LIMIT.

The name that you give to the GDG for the Automatic Log Save send data set must match the name that you give to `userid.gdname1` in BLMSDCS. The DCB information must match the DCB information for your Automatic Log Save send data set, which is derived from the DCB information for your SDLDS.

To create a GDG base for the SQL data set, modify and run BLGDG1 as follows:

1. On the BLGDG card, change `userid.dsname.gdg` to the name you want to use for the generation data group for the SQL data set.

2. On the BLGDG card, change the DCB values as follows:
   
   `DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)`

   **Note:** Use the appropriate BLKSIZE for your installation.

3. On the DEFINE statement, change `userid.dsname.gdg` to the name you want to use for the generation data group for the SQL data set. This name must match the name you used on the BLGDG card.

4. Change any other options in the DEFINE statement at your discretion. For example, you may want to increase or decrease the value for LIMIT.

The name that you give to the GDG for the SQL data set must match the name that you give to `userid.gdname2` in BLMSSND.

### Preparing the JCL

Copy and edit each of the following JCL members from the sample library, SBLMSAMP. Do not submit these JCL members after you update them. Some of these members are used by TSP BLGTDBXM and TSP BLGTDBX1. One of the JCL members, BLMSTEP2, you submit on the MVS image where DB2 resides to compile, link, and bind a new plan for DSNTEP2. See “Directions for Updating JCL” on page 13-25 for general directions on updating JCL.

**BLMSDSD** Runs on the MVS image where the Information/Management send database resides. This job is submitted:

- By you to initiate the DB2 Extract Facility
- By TSP BLGTDBXM if it completes successfully but does not build an SQL data set
- By JCL BLMSRCV if it updates the DB2 database successfully

This JCL does the following tasks:

- Starts TSP BLGTDBXM using Information/Management in batch mode.
- Sends a status message based on the success or failure of TSP BLGTDBXM.
BLMSDCS  Runs on the MVS image where the Information/Management send database resides. It is submitted by TSP BLGTDBXM and does the following tasks:

- Copies the Automatic Log Save send data set to a generation of a GDG.
- Deletes the Automatic Log Save send data set.

This job must complete before the next offload of the SDLDS. User exit BLGUT3EX allocates the send data set as a new data set. If the send data set already exists, BLGUT3EX encounters an allocation error.

**Note:** The deletion of the SQL data set and the Automatic Log Save send data set must occur before the next iteration of the DB2 Extract Send Utility.

BLMSSND  Runs on the MVS image where the Information/Management send database resides. It does the following tasks:

- Builds the job-stream (BLMSRCV) containing the SQL statements to be processed on the MVS image where DB2 resides. (The SQL statements were built by user exit BLMSSGEN from the data in the Automatic Log Save send data set.)
- Submits job BLMSRCV to the internal reader for routing to the MVS image where DB2 resides.
-Copies the SQL data set to a generation of a GDG and deletes the SQL data set.

BLMSRCV  A partial job that is combined with the in-stream SQL statements from BLMSSND and BLMSL1 to create the job-stream for submission to the internal reader. It does the following tasks:

- Deletes the SQL data set from the previous run.
- Adds the in-stream SQL statements from BLMSSND.
- Starts copying the in-stream SQL statements into a new data set for use by the DSNTEP2 program.

BLMSL1  A partial job that is appended to BLMSRCV after the in-stream SQL statements. It does the following tasks:

- Finishes copying the in-stream SQL statements into a new data set for use by the DSNTEP2 program.
- Runs the PL/I program DSNTEP2 to process the SQL statements in batch.
- Transmits a message to the system and user ID specified. The message that is sent is based on the return code from DSNTEP2 as follows:

  0000       Good message
  0004       Warning message
>0004 Failure message

- If DSNTEP2 returns a code of 0, submits job BLMSDSD for routing to the MVS image where the send database resides for execution.

BLMSL2 For recovery when errors occur after an SQL data set was successfully generated. This JCL runs on the MVS image where the DB2 database resides. For this job, you specify the input data set instead of using one that is automatically sent from the MVS system where Information/Management resides. It performs the following tasks:

- Runs the PL/I program DSNTEP2 to process the SQL statements in batch.

- Transmits a message to the system and user ID specified. The message that is sent is based on the return code from DSNTEP2 as follows:

  0000 Good message
  0004 Warning message
  >0004 Failure message

BLMSTEP2 For submission on the MVS image where DB2 resides to compile, link, and bind a new plan for DSNTEP2.

Directions for Updating JCL
1. Edit each member. Make your changes using uppercase text.
2. Follow the instructions in the prologue of each member, and printed in this chapter, to find the specific job steps you need to update.
3. Verify the accuracy of each change you make against your LOGSAVE record, TSP, and other JCL members.
4. Be sure to change the job card to your local standards.
5. Save the updated member.

Preparing BLMSDSD
1. Change the job card to your local standards. This job is routed to the MVS image where the Information/Management send database resides; you may need user and password information.
2. Change node on the /*ROUTE XEQ card to the node for the MVS image where the job is to run.
3. Change node.userid on the /*ROUTE PRINT card to the node and user ID that is to receive the job output.
4. In step SQLGEN, change:
   a. The data set names for the STEPLIB, ISPPROF, ISPPLIB, ISPTLIB, ISPMLIB, ISPSLIB, BLGTRACE, ISPLLIB, SYSPROC, and SYSTSPRT DD names to correspond to the data set names at your installation. Verify that the
Setting Up the DB2 Extract Facility

report format table concatenation contains the RDMTs needed by the DB2 Extract Send Utility.

b. The value for SESS(xx) to the session-parameters member of your send database.

5. In step GOODGEN, change:
   a. sample message to your message for a successful completion (return code = 0) of TSP BLGTDBXM.
   b. node.userid to the destination for the message.

6. In step FAILGEN, change:
   a. sample message to your message for an unsuccessful completion of TSP BLGTDBXM (return code ≠ 0).
   b. node.userid to the destination for the message.

Copy this JCL to a data set on the MVS image where the DB2 database resides.

Preparing BLMSDCS
1. Change the job card to your local standards.

2. In step SAVESEND, change:
   a. All occurrences of userid.dsname1 to the name of the Automatic Log Save send data set that is in the LOGSAVE record.
   b. userid.gdgname1 to the name of the GDG for the send data set.
   c. The DCB parameters as necessary to match those of the Automatic Log Save send data set.

Preparing BLMSSND
1. Change the job card to your local standards.

2. In steps BLDJOB, SAVESQL, and SAVEINFO, change userid.dsname2 to the SQL data set name specified in the LOGSAVE record.

3. On the SYSUT1 DD CARD, you may need to change the names of the two other data sets to the names of your local data sets.

4. In step SAVESQL, change userid.dsname2 to the SQL data set name specified in the LOGSAVE record. Put this same name in the DSNAME parameter of the DCB statement.

5. In step SAVESQL, change userid.gdgname2 to the name of the GDG for the SQL data set.
Propagating to a DB2 Database

Setting Up the DB2 Extract Facility

Preparing BLMSRCV

1. Change the job card to your local standards. This job is routed to the MVS image where the DB2 database resides; you may need user and password information.

2. Change node on the /*ROUTE XEQ card to the node for the MVS image where the job is to run.

3. Change node.userid on the /*ROUTE PRINT to the node and user ID that is to receive the job output.

4. Change userid.loadlib to the load library that contains DSNTEP2. If necessary, specify the PL/I system libraries.

5. In step DELETE, change userid.dsname2 to a name for the SQL data set that will be created on the MVS image where DB2 resides. This name must match the data set name on the SYSUT2 card in BLMSL1.

Preparing BLMSL1

1. In step SYSUT2, change userid.dsname2 to a name for the SQL data set. If the DB2 database is on the same MVS image as the Information/Management send database, this name must be different from the name you specified in the LOGSAVE record for the SQL data set. If they are the same, data will be overwritten.

2. You may need to change the UNIT and SPACE parameters according to your local standards and the volume of data being processed.

3. In step RUNSQL, change:
   a. userid.dsname2 to the SQL data set name you specified in the previous step.
   b. prefix_name to your prefix, the high-level qualifier for system resource data sets.
   c. db2_id to your DB2 subsystem ID.
   d. plan_name to your plan name for DSNTEP2.
   e. userid.loadlib to the load library that contains the executable load module, DSNTEP2.

4. In step GOODMSG, change:
   a. this is a sample message... to your message for a successful completion (return code = 0000) of the RUNSQL step.
   b. node.userid to the destination for this message.

5. In step WARNMSG, change:
   a. this is a sample message... to your message for a return code of 0004 from the RUNSQL step.
   b. node.userid to the destination for this message.

6. In step FAILMSG, change:
a. *this is a sample message...* to your message for an unsuccessful completion (return code > 0004) of the RUNSQL step.

b. `node.userid` to the destination for this message.

7. In step SUBDBXG, change `userid.jcl(BLMSDSD)` to the name of the data set that you copied BLMSDSD into on the MVS image where the DB2 database resides.

### Preparing BLMSL2

1. In step RUNSQL, change:

   a. `userid.dsname2` to the name of the data set that contains the SQL statements you want to process.
   
   b. `prefix_name` to your prefix, the high-level qualifier for system resource data sets.
   
   c. `db2_id` to your DB2 subsystem ID.
   
   d. `plan_name` to your plan name for DSNTEP2.
   
   e. `userid.loadlib` to the load library that contains the executable load module, DSNTEP2.

2. In step GOODMSG, change:

   a. *this is a sample message...* to your message for a successful completion (return code = 0000) of the RUNSQL step.
   
   b. `node.userid` to the destination for this message.

3. In step WARNMSG, change:

   a. *this is a sample message...* to your message for a return code of 0004 from the RUNSQL step.
   
   b. `node.userid` to the destination for this message.

4. In step FAILMSG, change:

   a. *this is a sample message...* to your message for an unsuccessful completion (return code > 0004) of the RUNSQL step.
   
   b. `node.userid` to the destination for this message.

### Preparing BLMSTEP2

1. Change the job card to your local standards.

2. In step JOBLIB CARD, add the PL/I system libraries, if necessary.

3. In step PPLI.SYSIN, change `userid.dnsamp` to the data set that contains the DSNTEP2 program.

4. In step PC.DBRMLIB, change `userid.dbrlib` to the name of your DBRM library.

5. In step PC.SYSLIB, change `userid.srclib` to the name of the data set for the modified source.

6. In step LKED.SYSLMOD, change `userid.runlib.load` to the name of the target load library.
7. In step PH01PS02, change:
   a. `userid.dbrmlib` to the name of your DBRM library.
   b. `plan_name` to your plan name.
   c. `userid.runlib.load` to your target load load library.

### Modifying the DB2 Extract TSP

Before you modify TSP `BLGTDBXM`, understand its logic. The logic of TSP `BLGTDBXM` is as follows:

1. Perform `BLMSSGEN` setup (passing a 1 in the **Literal/Test data** field). If an error occurs, issue a message and exit the TSP.

2. Set `TSCAUFLD` field to blanks. `TSCAUFLD` is used later to determine whether to submit JCL to copy the Automatic Log Save send data set to a generation of a GDG.

3. Call user exit `BLGUT4WT` to:
   a. Read information from the LOGSAVE record.
   b. Put the information into the TSCA variable data area.
   c. Wait for the time interval specified in the LOGSAVE record.

4. If `BLGUT4WT` returns an error:
   a. Issue a message.
   b. Perform `BLMSSGEN` cleanup (passing a 3 in the **Literal/Test data** field).
   c. Exit the TSP.

5. If `BLGUT4WT` finishes without an error, call user exit `BLGUT4EX` to offload the SDLDS to the Automatic Log Save send data set.

6. If `BLGUT4EX` returns an error:
   a. Issue a message.
   b. Perform `BLMSSGEN` cleanup (passing a 3 in the **Literal/Test data** field).
   c. Exit the TSP.

7. If `BLGUT4EX` finishes without an error:
   a. Set `TSCAUFLD` to OFFLDOK
   b. Perform `BLMSSGEN` extract (passing a 2 in the **Literal/Test data** field).

8. If `BLMSSGEN` finishes with a TSCAFRET that is not 0 or a TSCAFRES of 12:
   a. Issue a message.
   b. Perform `BLMSSGEN` cleanup (passing a 3 in the **Literal/Test data** field).
   c. Submit JCL `BLMSDCS` to copy the Automatic Log Save send data set to a generation of a GDG.
   d. Exit the TSP.

9. If `BLMSSGEN` finishes with a TSCAFRES of 4 or 8, no SQL data set was built; therefore, the JCL to process the SQL is not submitted.
   a. Submit JCL `BLMSDSD` to start the next send.
   b. Issue a message indicating the DB2 Extract Send Utility was successful.
c. Perform BLMESSGEN cleanup (passing a 3 in the **Literal/Test data** field).

d. Submit JCL BLMSDCS to copy the Automatic Log Save send data set to a generation of a GDG.

e. Exit the TSP.

10. If BLMESSGEN finishes without an error and SQL was generated, submit JCL BL MSSND to process the SQL data set.

The JCL builds a job stream to load the SQL data into DB2 and routes it to the MVS image where the DB2 database resides. It also copies the SQL data set to a member of a GDG and deletes the data set. The SQL data set must be deleted before the next time interval elapses because BLMESSGEN attempts to create the data set; therefore, the data set must not exist.

11. If there is an error submitting the JCL:

   a. Issue a message.
   
   b. Perform BLMESSGEN cleanup (passing a 3 in the **Literal/Test data** field).
   
   c. Submit JCL BLMSDCS to copy the Automatic Log Save send data set to a generation of a GDG.
   
   d. Exit the TSP.

12. If there is no error submitting the JCL:

   a. Issue a message indicating the DB2 Extract Send Utility was successful.
   
   b. Perform BLMESSGEN cleanup (passing a 3 in the **Literal/Test data** field).
   
   c. Submit JCL BLMSDCS to copy the Automatic Log Save send data set to a generation of a GDG.
   
   d. Exit the TSP.

Using PMF, update the following statements in the BLGTDWXM TSP as follows:

1. Change BLM.VXRXM.SBLMSAMP(BLMSDSD) to the name of your library that contains the JCL you modified in “Preparing BLMSDSD” on page 13-25. (This library is on the MVS image where the send database resides.)

2. Change BLM.VXRXM.SBLMSAMP(BLMSSND) to the name of your library that contains the JCL you modified in “Preparing BLMSSND” on page 13-26. (This library is on the MVS image where the send database resides.)

3. Change BLM6.VXRXM.SBLMSAMP(BLMSDCS) to the name of your library that contains the JCL you modified in “Preparing BLMSDCS” on page 13-26. (This library is on the MVS image where the send database resides.)
This panel shows only the lines in the BLGTDBXM TSP that you change to propagate data to DB2.

<table>
<thead>
<tr>
<th>Function Label</th>
<th>Literal</th>
<th>Data</th>
<th>Get Apply</th>
<th>Field</th>
<th>Name</th>
<th>Field Name</th>
<th>Not Field Name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MOVEVAR</strong></td>
<td>SELECT CMD(SUBMIT)</td>
<td>'BLM.VxRxMx.SBLMSAMP(BLMSDSD)'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MOVEVAR</strong></td>
<td>SELECT CMD(SUBMIT)</td>
<td>'BLM.VxRxMx.SBLMSAMP(BLMSND)'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MOVEVAR</strong></td>
<td>SELECT CMD(SUBMIT)</td>
<td>'BLM.VxRxMx.SBLMSAMP(BLMSDCS)'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MOVEVAR</strong></td>
<td>SELECT CMD(SUBMIT)</td>
<td>'BLM.VxRxMx.SBLMSAMP(BLMSDCS)'</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you want to change the maximum number of records to offload each time an enqueue is obtained on the SDLDS, you can change the RECS= and %FULL= parameters in the control line that calls user exit BLGUT4EX. As shipped, the BLGTDBXM TSP has RECS=20. See “Modifying the BLGDUMP1 TSP” on page 12-20 for an example of how to change the control line.

File the updated TSP. After you file the updated TSP, you must copy it from the write panel data set to a read panel data set so it is available for use. After you do that, you are ready to start the batch job that runs the TSP if you have completed all the other setup steps.

**Starting Send and Update Processing**

To start normal Send and Update processing, start the DB2 Extract Facility by submitting JCL BLMSDSD on the MVS image where the send database resides. Before you submit JCL BLMSDSD, verify that:

- The set up for your send database is complete.
- The set up for the DB2 database is complete.
- The status of the send LOGSAVE record is enabled.
Example of Propagating Problem Data to DB2

This section illustrates Information/Management data propagated to DB2.

This is one panel of an Information/Management problem record. Figure 19 on page 13-12 maps some of the fields on this panel to DB2 tables PROBLEM_DATA and PROBLEM_TEXT.

The DB2 table in Figure 20 shows data from this record propagated to DB2 table PROBLEM_DATA. It shows two sample records with 8 fields. The ninth field is a freeform text field, so it is mapped to another table, which is illustrated in Figure 21 on page 13-33.

---

<table>
<thead>
<tr>
<th>RNID</th>
<th>REPORTER_NAME</th>
<th>PROBLEM_TYPE</th>
<th>REPORTER_DEPT</th>
<th>REPORTER_STAT</th>
<th>REPORTER_PHONE</th>
<th>INITIAL_PRIORITY</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001263</td>
<td>EMERSON</td>
<td>PERFORM</td>
<td>T48B</td>
<td>INITIAL</td>
<td>555-3333</td>
<td>03</td>
<td>THIS IS A BAD PROBLEM JJJPROB</td>
</tr>
<tr>
<td>00008900</td>
<td>THOREAU</td>
<td>HARDWARE</td>
<td>T44A</td>
<td>OPEN</td>
<td>7579</td>
<td>02</td>
<td>SEND</td>
</tr>
</tbody>
</table>

Figure 20. Propagated Data Retrieved from DB2
You might have used these SQL statements to create the DB2 table and index:

```
CREATE TABLE authorization_id.PROBLEM_DATA
  (RNID CHAR(08) NOT NULL ,
   REPORTER_NAME CHAR(15) ,
   PROBLEM_TYPE CHAR(08) ,
   REPORTER_DEPT CHAR(11) ,
   REPORTER_STAT CHAR(07) ,
   REPORTER_PHONE CHAR(13) ,
   INITIAL_PRIORITY CHAR(02) ,
   DESCRIPTION CHAR(45) ,
   PRIMARY KEY (RNID)
) IN database_name.tablespace_name;
CREATE UNIQUE INDEX authorization_id.index_name
ON authorization_id.PROBLEM_DATA (RNID);
```

This example shows freeform text in Information/Management. The RDMT in Figure 19 on page 13-12 maps it to DB2 table PROBLEM_TEXT. Figure 21 shows how this data could be propagated to DB2.

```
BLGITDES DESCRIPTION TEXT LINE 1 OF 16
** 08/23/93 This is an example of freeform text
** 08/23/93 This is another example of freeform text

**** *********** BOTTOM OF DATA ***
```

```
RNID  SEQ_NUM TEXT
-------  -------- -----------------------------------
00001263 00000001 This is an example of freeform text
00001263 00000002 This is another example of freeform text
```

Figure 21. Propagated Freeform Text Retrieved from DB2
You might have used these SQL statements to create the DB2 table and index:

```
CREATE TABLE authorization_id.PROBLEM_TEXT
  (RNID CHAR(/zerodot8) NOT NULL ,
   SEQ_NUM CHAR(/zerodot8) ,
   TEXT CHAR(250) ,
   PRIMARY KEY (RNID,SEQ_NUM) )
IN database_name.tablespace_name;
CREATE UNIQUE INDEX authorization_id.index_name
ON authorization_id.PROBLEM_TEXT (RNID,SEQ_NUM);
```

### Stopping the DB2 Extract Facility

To stop the DB2 Extract Facility, toggle the status of the LOGSAVE record from `enabled` to `disabled`. You can use the following IRC from the Primary Options Menu to toggle the status:

```
update r logsave,4,9
```

When you stop the send utility, the update utility also stops. If a send is currently in progress (that is BLMSDSD has been started and TSP BLGTDBXM has called user exit BLGUT4WT and the exit is waiting), the current send finishes waiting. Then the current send offloads the SDLDS, builds the SQL data set, and routes BLMSRCV for execution. The update utility (BLMSRCV) executes and submits BLMSDSD on the MVS image where the send database resides. BLMSDSD fails in step OFFLD because of a TSP error indicating that the status of the LOGSAVE record is disabled.

### Disabling Automatically Starting the Next Send

If you want to disable automatically starting the next send after a successful update, delete the step that submits the JCL to start the next send from the JCL for the DB2 Update Utility. You also need to modify the TSP so that it does not submit the JCL to start a send if no SQL is built.

To implement the DB2 Extract Facility so that it does just one send and one update, then stops, modify the TSP and JCL as follows:

1. **Using PMF, copy and update TSP BLGTDBXM as follows:**
   a. Insert a control line to branch to label ISSUGMSG after the control line that tests whether an SQL data set was built. This disables the TSP control lines that submit JCL to perform the next send if no SQL is built.
   b. Change the line of the TSP that submits JCL BLMSSND to submit the new version of BLMSSND that you copy or update according to step 2c.
This panel shows where you need to insert the control line to branch to label ISSUGMSG and change the line that submits the JCL.

<table>
<thead>
<tr>
<th>BLMITUCU</th>
<th>FUNCTION LINE SUMMARY</th>
<th>LINE n OF nn</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTION</td>
<td>LABEL</td>
<td>LITERAL</td>
</tr>
<tr>
<td>NAME</td>
<td>NAME</td>
<td>DATA</td>
</tr>
<tr>
<td>'I</td>
<td>TESTFIELD SQLBUILT 0</td>
<td>NO</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>**MOVEVAR</td>
<td>SELECT CMD(SUBMIT</td>
<td>'BLM.VxRxMx.SBLMSAMP(BLMSSND)'</td>
</tr>
<tr>
<td>**MOVEVAR</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>

2. Copy and update the following JCL members:
   a. BLMSDSD. In step SQLGEN, change TSP BLGTDBXM to the name you used for the TSP you modified according to step 1 on page 13-34.
   b. BLMSL1. Delete step SUBDBXG.
   c. BLMSSND. In step BLDJOB, change BLMSL1 to the name of the copy of BLMSL1 you modified in step 2b.

Each time you submit the modified BLMSDSD, one send and one update utility is performed. JCL to start another DB2 Send Utility is not submitted.

**Propagating Data to Both Information/Management and DB2**

If you want to propagate data to both Information/Management and DB2, set up the Automatic Log Save Facility as described in “Setting Up the Automatic Log Save Send Function” on page 12-6 for data propagation to both Information/Management and DB2. You also need to modify TSP BLGTDBXM and the JCL for both the Automatic Log Save Facility and the DB2 Extract Facility as described in this section.

The changes alter the JCL for the Automatic Log Save Facility to use the DB2 Extract TSP so that both a send data set and an SQL data set are created. Because the JCL for the Automatic Log Save Receive Function submits JCL to start the next send after a successful receive, modify the JCL for the DB2 Update Utility so that it does not submit JCL to start a send after a successful update.
When you use the Automatic Log Save and DB2 Extract Facilities with these changes, the Automatic Log Save Facility drives the process, and an Automatic Log Save Receive failure causes both facilities to stop. After the send and SQL data sets are created, JCL is submitted by TSP BLGTDBXM to process the SQL data set and start the DB2 Update Utility. If the DB2 Update Utility fails, the failure has no effect on Automatic Log Save Send and DB2 Extract Send processing.

**Changing the TSP**

Change TSP BLGTDBXM as described in “Modifying the DB2 Extract TSP” on page 13-29. In addition, make the following changes:

1. Delete the control line that does a SETFIELD of TSCAUFLD to OFFLDOK. This disables the submission of the JCL to copy the send data set to a GDG because BLMSASD or BLMSASDA will do the copying.

2. Insert a control line to branch to label ISSUGMSG after the control line shown in this panel. This disables the TSP control lines that submit JCL to perform the next send if no SQL is built.

3. Change the line that submits JCL BLMSSND to submit the new version of BLMSSND that you copy and modify according to step 2 under “Changing the JCL for the DB2 Extract Facility” on page 13-37.

This panel shows the line you need to delete. This panel also shows where you need to insert the control line to branch to label ISSUGMSG and change the line that submits the JCL.

<table>
<thead>
<tr>
<th>BLMITUCU</th>
<th>FUNCTION LINE SUMMARY</th>
<th>LINE n OF nn</th>
</tr>
</thead>
<tbody>
<tr>
<td>FUNCTION NAME</td>
<td>LABEL NAME</td>
<td>LITERAL DATA</td>
</tr>
<tr>
<td>'d SETFIELD</td>
<td>NO</td>
<td>TSCAUFLD</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>'i TESTFIELD SQLBUILT 0</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>'''' MOVEVAR SELECT CMD(SUBMIT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'''' MOVEVAR 'BLM.VxRxMx.SBLMSAMP(BLMSSND)'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
</tbody>
</table>
Changing the JCL for the Automatic Log Save Facility

To enable data propagation to both an Information/Management receive database and a DB2 database, make the following additional modifications to the JCL members for the Automatic Log Save Facility:

1. BLMSAL1. In step SUBSND, refer to the version of JCL BLMSASDA that you modified according to step 2 in this section and that is on the MVS image where the receive database resides.

2. BLMSASD and BLMSASDA
   a. In step OFFLD, change TSP(BLGTDUMP1) to refer to the version of TSP BLGTDBXM you copied and modified according to “Changing the TSP” on page 13-36. Modify data set names as required; for example, verify that the RFT data set concatenation contains the RDMTs required for the DB2 Extract Send Utility.
   b. In step GOODOFF, change the GOODOFF DD contents if necessary. This message is sent for a successful completion (return code = 0000) from step OFFLD.
   c. In step FAILOFF, change the FAILOFF DD contents if necessary. This message is sent for an unsuccessful completion (return code ≠ 0000) from step OFFLD.
   d. In step SUBSND, change the data set name on IEFRDER to refer to the copied and modified JCL BLMSASDA.
   e. In step COPY2, change the data set name to refer to the version of JCL BLMSAL1 you modified in step 1 in this section.

Changing the JCL for the DB2 Extract Facility

Make the following additional modifications to the JCL members for the DB2 Extract Facility:

1. BLMSL1. Delete step SUBDBXG so that the JCL for the next send is not submitted.

2. BLMSSND. In step BLDJOB, change BLMSL1 to refer to the version of BLMSL1 you modified in step 1 of this section.

Starting Propagation to Both

To start data propagation to both an Information/Management receive database and a DB2 database, on the MVS image where the send database resides, submit the version of BLMSASD you modified in step 2 under “Changing the JCL for the DB2 Extract Facility.”
DSNTEP2 Messages

DSNTEP2 writes messages to SYSPRINT. When the RUNSQL step of job BLMSRCV finishes, the job output, which includes the messages from DSNTEP2 if SYSPRINT is not allocated to a data set, is sent to the user ID specified on the ROUTE PRINT card in the JCL.

Refer to the prologue of the DSNTEP2 program for a list of the messages it returns.

Recovering from Errors

The DB2 Extract Facility provides the following tools to assist you in recovering from errors:

- TSP BLGTDBX1 parses an Automatic Log Save send data set into SQL when the send data set was left unparsed because the TSP BLGTDBXM stopped. If the send data set was successfully copied to a GDG and the send data set was deleted, you must manually create the send data set and copy the most recent generation of the GDG into the send data set before running TSP BLGTDBX1.

- The Automatic Log Save send data set and the SQL data set are copied to GDGs on the MVS image where the Information/Management send database resides.

- A synchronization table in DB2 and a synchronization counter in the Information/Management SDIDS are provided to store the sequence number. The first SQL statement in the SQL data set checks the sequence number and issues a 0 divide error if the number is not the one that was expected. DB2 Update Utility returns an error message and does not submit the JCL to start the next DB2 Send Utility. If the sequence number in the DB2 SYNCH_TABLE is the one expected, the rest of the SQL statements are processed. The last SQL statement in the SQL data set updates the SYNCH_TABLE sequence number. Then the DB2 Update Utility submits the JCL to start the next DB2 Send Utility. See “Maintaining Synchronization” on page 11-22 for more information about the synchronization counters and how to determine their values.

- BLMSL2 is a batch job that processes the SQL statements in the SQL data set. When the DB2 Update Utility stops, the MVS image where the Information/Management send database resides may have one or more SQL data sets saved in GDGs. Under normal processing, there is only one unprocessed SQL data set. BLMSL2 is a batch interface to DSNTEP2. It runs on the MVS image where DB2 resides to allow these data sets to be processed into DB2. After the data sets have been processed, you can restart the DB2 Extract Send Utility.
Recovering from a BLGTDBXM Failure

When TSP BLGTDBXM fails, follow this procedure to recover:

1. Refer to the messages written to SYSPRINT on the MVS image where Information/Management resides and to the output from BLMRSCV to diagnose problem:
   a. Examine the return and reason codes from the user exit that failed.
   b. Look up the return and reason codes in the table for the user exit:
      - For user exit BLGUT4EX, see Table 38 on page 14-7.
      - For user exit BLGUT4WT, see Table 39 on page 14-8.
      - For user exit BLMSSGEN, see Table 40 on page 14-12.
      The description column explains the source of the error and indicates what to do to correct the problem.
   c. Examine the job output. Consider these questions:
      - Was the job submission to the internal reader on the send database successful? Check step SUBJOB in BLMSSND.
      - Was the job routed to the correct MVS image where your DB2 database resides? Check the /* ROUTE XEQ card in BLMSRCV.
      - Are the attributes of the SQL data set correct in the LOGSAVE record on the send database?

2. If the SQL data set was created but not processed:
   a. Manually copy it to a GDG.
   b. Transmit the SQL data set to the MVS image where the DB2 database resides and receive the data set.
   c. Prepare and submit BLMSL2 as described in “Preparing BLMSL2” on page 13-28 to process the SQL data set to update DB2 in batch.

3. If the SQL data set was not created:
   a. Determine why it was not created and fix the problem.
   b. Change the Time interval in minutes field in the LOGSAVE record to 0.
   c. If the send data set was copied to a GDG and the send data set was deleted, create the send data set and copy the most recent generation of the send GDG into it.
   d. Run TSP BLGTDBX1 to reprocess the Automatic Log Save send data set.
   e. After BLGTDBXM1 completes successfully, change the Time interval in minutes field in the LOGSAVE record back to its previous value.
   f. Delete the send data set if you had to create it.

Recovering When There Is a Receive Error
If the SQL data set was generated, but the receiving system encountered an error, or the receive JCL was not submitted, do the following tasks:

1. Determine the cause of the problem. Consider the questions in step 1c on page 13-39 in “Recovering from a BLGDBXM Failure” on page 13-39 plus the following questions:
   - Do you have the DB2 tables built with the same authorization ID that is specified in the LOGSAVE record on the send database?
   - Is your DB2 environment active?

2. If the problem occurred with DB2, for example the system was down, you can resubmit the SQL data sets with BLMSL2 after the problem has been fixed.

3. If the error was caused by an incorrect RDMT:
   a. Use BLGUT8 to rebuild the RDMT.
   b. Use TSP BLGTDBX1 to regenerate the SQL data set. Copy the generation of the send data set to be processed into the send data set and run BLGTDBX1 to regenerate the SQL data set.
   c. Prepare and submit BLMSL2 as described in “Preparing BLMSL2” on page 13-28 to process the SQL data set to update DB2 in batch.


Recovering When the SDLDS Is Full
When the SDLDS is full, you cannot update the LOGSAVE record. If the information in the LOGSAVE record (for example, the Time interval in minutes is too long, or the Status is disabled) does not allow for immediate offload using TSP BLGDBXM, you can use the following steps to manually start DB2 Extract Send and Update processing:

1. Manually allocate the send data set using the name and data set characteristics specified in the LOGSAVE record.
2. Use BLGUT4 to offload the SDLDS to the send data set allocated in step 1.
3. Run TSP BLGTDBX1 to generate the SQL and start DB2 Update processing.
4. Submit JCL BLMSDCS to copy the send data set to a GDG and delete the send data set.

If you are also using the Automatic Log Save Facility to propagate data to a receive database, use BLGUT3 to load the records from the send data set into the receive database. Refer to the Operation and Maintenance Reference for information on using BLGUT3.
Resetting Synchronization

If you encounter a synchronization error from the DB2 Update Utility, one or more SQL data sets have not been processed. This could happen if:

- SQL is successfully generated, but it is not used to update DB2 because of an error.
- You start the DB2 Extract Send Utility again without first successfully processing the first SQL data set.
- The synchronization error occurs and you have two SQL data set to process.

Reset Synchronization Scenario 1

Synchronization is lost. Copies of the data sets that have not been received are available.

1. Determine what caused the loss of synchronization.
2. If the problem occurred with DB2, for example the system was down, you can resubmit the SQL data sets with BLMSL2 after the problem is fixed. See “Preparing BLMSL2” on page 13-28 for more information about BLMSL2.

Reset Synchronization Scenario 2

The scenario in “Reset Synchronization Scenario 1” depends on recovering the Automatic Log Save send data sets in order by the sequence number so that you can resubmit them to DB2. If the data sets are not available:

1. Rebuild the DB2 database using the procedure currently in place at your data center for loading Information/Management data into DB2.
2. Determine the sequence number stored in the SDIDS of the Information/Management database. See “Maintaining Synchronization” on page 11-22 for information on how to do this.
3. Reset the synchronization count in DB2 table SYNCH_TABLE to equal the one in the Information/Management send database’s SDIDS. You can modify and use the following SQL statement to do this:

   ```sql
   UPDATE authorization_id.SYNCH_TABLE
   SET SYNCH_NUM = new_value
   WHERE SYNCH_NUM = current_value;
   ```

   Do not use the INSERT statement because it would add another row to the table. This table must have only one row.
Data Propagation User Exits

This chapter contains Product-sensitive Programming Interface and Associated Guidance Information.

The user exits described in this section are intended for use by the Automatic Log Save Facility and DB2 Extract Facility. After you understand these exits and how they work together, you can use them for your own purposes.

User exits BLGUT3EX and BLGUT4EX receive their input from BLGUT3WT and BLGUT4WT respectively. User exits BLGUT3WT and BLGUT4WT receive their input from the LOGSAVE record in the Information/Management database. User exit BLMSSGEN receives its input from the LOGSAVE record, relational data mapping tables (RDMTs), and BLGUT4EX. The data is passed through the TSCA variable data area. Refer to the Terminal Simulator Guide and Reference for information on the TSCA variable data area.

Note: User exits which require that information be specified on the TSP USEREXIT control line cannot be run from a TSX. Refer to the Environment section of each of the user exits in this chapter to determine whether or not the user exit can be run from a TSX.

User Exits

BLGUT3EX—Online Recovery Utility

Loads the Information/Management database with records that were offloaded by the BLGUT4EX user exit.

Input

TSCA variable data area output by BLGUT3WT.
Send data set that was output by BLGUT4EX and received as the receive data set.

Output

Records from the receive data set converted to SDDS records.

Environment

TSP and TSX.
A return code and reason code as listed in Table 36 on page 14-2. The user exit writes informational and error messages to SYSPRINT. If you allocate SYSPRINT to a data set, the record format must be VBA.
<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>Error allocating the receive data set. Check the data set name. Make sure the data set was received. Send it again.</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>Logical record skipped because the data is not valid. Message BLG03906 was written. Update the record on the send database so that it will be sent again.</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>Freeform text from one or more records cannot be processed. Message BLG21305 was written. Update the record on the send database so that it will be sent again.</td>
</tr>
<tr>
<td>4</td>
<td>17</td>
<td>Records are not in sequential order.</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>Internal control blocks were not found.</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>Error opening the receive data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>Error reading records from the receive data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>Error opening the SDIDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>Error closing the SDIDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>Error updating the SDIDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>Error reading the SDIDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>Error closing the receive data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>Error freeing the receive data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>Error cognizing the records added to the SDDS.</td>
</tr>
</tbody>
</table>
The BLGLOAD1 TSP contains an example of using this user exit.

BLGUT3WT–Initialize for Receive

Initializes the TSCA variable data area and waits for the interval specified in the LOGSAVE record.

**Input**
- LOGSAVE record for a receive database.

**Output**
- TSCA variable data area.

**Environment**
- TSP and TSX.
- A return code and reason code as listed in Table 37. The user exit writes informational and error messages to SYSPRINT. If you allocate SYSPRINT to a data set, the record format must be VBA.

---

### Table 36 (Page 2 of 2). BLGUT3EX Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>20</td>
<td>The sequence number in the send data set does not match the sequence number in the receive database’s SDIDS. Synchronization is lost. See “Restoring Synchronization” on page 12-37 for information on how to restore synchronization.</td>
</tr>
</tbody>
</table>

---

### Table 37 (Page 1 of 4). BLGUT3WT Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>A database error occurred reading the LOGSAVE record.</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>The LOGSAVE record was not found. Make sure it exists.</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>An enqueue error occurred reading the LOGSAVE record. Try the process again.</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>The database contains more than one LOGSAVE record. Delete one.</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>Not enough storage to read the LOGSAVE record. Check the region size.</td>
</tr>
<tr>
<td>Return Code (TSCAFRET)</td>
<td>Reason Code (TSCAFRES)</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>The LOGSAVE record is busy. Try again, or run DBCLEANUP.</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>Cannot read the LOGSAVE record. The record is damaged. Delete it and recreate it.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>Internal control blocks were not found.</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>The status of the LOGSAVE record is disabled. Enable it.</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>Cannot locate database 5. Check that the correct session-parameters member is being used.</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>Record with RNID LOGSAVE is not LOGSAVE type. Delete it and create the LOGSAVE record using the procedures described in Propagating to an Information/Management Database.</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>The <strong>Data set name</strong> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>12</td>
<td>The <strong>Block size</strong> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>The <strong>Logical record length</strong> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
</tbody>
</table>
### Table 37 (Page 3 of 4). BLGUT3WT Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>20</td>
<td>The <strong>Primary allocation blocks</strong> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>The <strong>Secondary allocation blocks</strong> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>36</td>
<td>The <strong>Unit type</strong> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>56</td>
<td>The <strong>Synchronize Send and Receive?</strong> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>60</td>
<td>The <strong>ABEND disposition</strong> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>68</td>
<td>The <strong>Receive filter data</strong> field was not found in the LOGSAVE record when a <strong>Receive filter prefix</strong> field was. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>A numeric field is not all numbers.</td>
</tr>
</tbody>
</table>
The BLGLOAD1 TSP contains an example of using this user exit.

**BLGUT4EX–Offload a Recovery Data Set**

Offloads the recovery data set (SDLDS) into a send data set.

**Input**

- TSCA variable data area output by BLGUT4WT.
- The SDLDS.

Optional parameters RECS= and %FULL= in the Literal/Test data field on the Data Field Specification panel of the calling USEREXIT control line.

- **RECS=nn** An optional parameter that specifies the number of physical records to be processed each time an enqueue is obtained on the SDLDS before releasing the enqueue. *nn* can range from 0 to 99999999. A value of 0 indicates that the SDLDS enqueue is not to be released until the offload is complete. The default value is 0.

- **%FULL=xxx** An optional parameter that specifies if, during offload processing, the SDLDS becomes more than *xxx* percent full, processing is to continue without releasing the enqueue on the SDLDS until the offload is complete. *xxx* can range from 0 to 100. The default value is 80. If RECS=nn is omitted or has a value of 0, then %FULL= is ignored.

**Output**

- A send data set.

**Environment**

- TSP only.

A return code and reason code as listed in Table 38 on page 14-7. The user exit writes informational and error messages to SYSPRINT. If you allocate SYSPRINT to a data set, the record format must be VBA.

---

### Table 37 (Page 4 of 4). BLGUT3WT Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>8</td>
<td>The LOGSAVE record has <em>send</em> in the Database type field and this user exit performs receive processing. Correct the record.</td>
</tr>
<tr>
<td>Return Code (TSCAFRET)</td>
<td>Reason Code (TSCAFRES)</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>Internal control blocks were not found.</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>Error opening the recovery data set (SDLDS). Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>Error allocating the send data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>Error opening the send data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>Error performing VSAM set on the SDLDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>Error getting an SDLDS record. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>SDLDS record is not valid. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>Error writing to the send data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>Error updating the SDLDS control record. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>Error reading the SDLDS control record. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>Error closing the SDLDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>Error closing the send data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>Error freeing the send data set. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>No SDLDS found. Look for other messages. Check whether you are using the correct session-parameters member.</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>Error opening the SDIDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>16</td>
<td>Error closing the SDIDS. Look for other messages.</td>
</tr>
</tbody>
</table>
The BLGDUMP1, BLGTDDBXM, and BLGTDDBXM1 TSPs contain examples of using this user exit.

**BLGUT4WT–Initialize for Send**

Initializes the TSCA variable data area and waits for the interval specified in the LOGSAVE record.

**Input**

LOGSAVE record for a send database.

**Output**

TSCA variable data area.

**Environment**

TSP and TSX.

A return code and reason code as listed in Table 39. The user exit writes informational and error messages to SYSPRINT. If you allocate SYSPRINT to a data set, the record format must be VBA.

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>17</td>
<td>Error reading the SDIDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>18</td>
<td>Error writing to the SDIDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>19</td>
<td>Error performing VSAM set on the SDIDS. Look for other messages.</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>Error processing optional parameters. Look for other messages.</td>
</tr>
</tbody>
</table>

Table 38 (Page 2 of 2). BLGUT4EX Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>An error occurred reading the LOGSAVE record.</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>The LOGSAVE record was not found. Make sure it exists.</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>An enqueue error occurred reading the LOGSAVE record. Try the process again.</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>The database contains more than one LOGSAVE record. Delete one.</td>
</tr>
</tbody>
</table>

Table 39 (Page 1 of 3). BLGUT4WT Return and Reason Codes
<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>20</td>
<td>Not enough storage to read the LOGSAVE record. Check the region size.</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>The LOGSAVE record is busy. Try again or run DBCLEANUP.</td>
</tr>
<tr>
<td>4</td>
<td>28</td>
<td>Cannot read the LOGSAVE record. The record is damaged. Delete the record and recreate it.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>Internal control blocks were not found.</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>The status of the LOGSAVE record is disabled. Enable it.</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>Cannot locate database 5. Check that the correct session-parameters member is being used.</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>Record with RNID LOGSAVE is not LOGSAVE type.</td>
</tr>
<tr>
<td>20</td>
<td>12</td>
<td>The Block size field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>The Logical record length field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>The Primary allocation blocks field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
</tbody>
</table>
Table 39 (Page 3 of 3). BLGUT4WT Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>24</td>
<td>The <em>Secondary allocation blocks</em> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>The <em>Unit type</em> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>The <em>Synchronize Send and Receive?</em> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>The <em>ABEND disposition</em> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>The <em>Data set name</em> field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged. Verify and correct the missing information.</td>
</tr>
<tr>
<td>24</td>
<td>4</td>
<td>A numeric field is not all numbers.</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>The LOGSAVE record has <em>receive</em> in the <em>Database type</em> field and this user exit performs send processing.</td>
</tr>
</tbody>
</table>

The BLGDUMP1 and BLGTDBXM TSPs contain examples of using this user exit. Correct the record.
Data Propagation User Exits

BLMSSGEN–SQL Setup, Extract, and Cleanup

Converts Information/Management records in an Automatic Log Save send data set format to SQL statements for propagating the Information/Management data to a DB2 database.

The user exit has three modes: setup, extract, and cleanup. For maximum performance, do setup and cleanup once for multiple calls to BLMSSGEN extract. The user exit writes informational and error messages to SYSPRINT. If you allocate SYSPRINT to a data set, the record format must be VBA.

Setup

Reads the LOGSAVE record and the RDMTs that are identified in the LOGSAVE record and stored in a data set in the report format table concatenation. Allocates storage for extract processing.

Input

A value of 1 in the Literal/Test data field on the Data Field Specification panel of the calling USEREXIT control line.

LOGSAVE record defined for DB2 Extract processing.

Report format table data set concatenation containing the RDMTs specified in the LOGSAVE record.

Output

BLMSSGEN environment established.

Environment

TSP only.

If an error occurs reading or validating an RDMT, TSCAUFLD is set to the name of the RDMT in error.

A return and reason code as listed in Table 40 on page 14-12.

Extract

Reads Information/Management records from the send data set identified in the LOGSAVE record and generated by user exit BLGUT4EX.

For each record read from the send data set, attempts to find the record type s-word identified in each RDMT. Uses the first RDMT that matches.

If no RDMTs match the Information/Management record, continues to the next record.

For each field contained in the RDMT that matches a field in the Information/Management record, takes the field data and generates SQL statements in the SQL data set.
Input
A value of 2 in the **Literal/Test data** field on the Data Field Specification panel of the calling USEREXIT control line.

TSCAVDAP pointing to data created by user exit BLGUT4WT.

Environment set up by BLMSSGEN setup.

LOGSAVE record defined for DB2 Extract processing.

Send data set that was output by BLGUT4EX.

Output
SQL data set.

Environment
TSP only.

A return and reason code as listed in Table 40.

Cleanup
Frees storage obtained during BLMSSGEN setup.

Input
A value of 3 in the **Literal/Test data** field on the Data Field Specification panel of the calling USEREXIT control line.

Environment set up by BLMSSGEN setup.

Output
BLMSSGEN environment freed.

Environment
TSP only.

A return and reason code as listed in Table 40.

The BLGTDBXM and BLGTDBXM1 TSPs contain examples of using this user exit.

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Successful completion. SQL statements generated.</td>
</tr>
<tr>
<td>0</td>
<td>4</td>
<td>No records were in the send data set. No SQL statements generated.</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>No SQL statements generated, including the UPDATE statements for synchronization. All records skipped. No processing errors.</td>
</tr>
<tr>
<td>Return Code (TSCAFRET)</td>
<td>Reason Code (TSCAFRES)</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>12</td>
<td>At least one record skipped because of processing errors. SQL statements generated unless all records were skipped.</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Incorrect or null <strong>Literal/Test data</strong> field. One character must be specified in the <strong>Literal/Test data</strong> field of the calling USEREXIT control line and it must be a value of 1, 2, or 3.</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>Performing extract processing and TSCAVDAL is not equal to the length set by user exit BLGUT4WT.</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>Performing setup processing and the environment is already established.</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>BLMSSGEN environment not established. Perform setup before performing extract or cleanup.</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>The API is active. BLMSSGEN cannot be called as part of an API transaction.</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>Out of storage.</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>Internal control blocks were not found. Contact IBM.</td>
</tr>
<tr>
<td>16</td>
<td>8</td>
<td>Internal error detected. Contact IBM.</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>BLMSSGEN environment damaged. Contact IBM.</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>RFT data set not specified in the session-parameters member.</td>
</tr>
<tr>
<td>20</td>
<td>8</td>
<td>Error allocating the RFT data set. Look for other messages.</td>
</tr>
<tr>
<td>20</td>
<td>12</td>
<td>Error opening the RFT data set. Look for other messages.</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>Error closing the RFT data set. Look for other messages.</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>Record format of RFT data set is not fixed.</td>
</tr>
<tr>
<td>Return Code (TSCAFRET)</td>
<td>Reason Code (TSCAFRES)</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>RDMT was not found in the RFT data set. TSCAUFLD contains the name of the RDMT.</td>
</tr>
<tr>
<td>20</td>
<td>28</td>
<td>Error reading RDMT from RFT data set. TSCAUFLD contains the name of the RDMT.</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>RDMT is not valid or is not of correct type. For example, it is not an RDMT, the name stored in the RDMT header does not match the name of the RFT data set member, or the RDMT is truncated. TSCAUFLD contains the name of the RDMT. Verify that name of the RFT data set member matches the name you used on the TABLE statement when you ran BLGUT8 to build the RDMT. Verify that the name you put in the RDMT name field of the LOGSAVE record is correct.</td>
</tr>
<tr>
<td>20</td>
<td>36</td>
<td>No record type s-word found in the RDMT. TSCAUFLD contains the name of the RDMT.</td>
</tr>
<tr>
<td>20</td>
<td>48</td>
<td>Logical record length of RFT data set is not 80.</td>
</tr>
<tr>
<td>20</td>
<td>52</td>
<td>Unknown error allocating the RFT data set, or an unknown error reading the RDMT. Verify that your session-parameters member allocates an RFT data set that contains RDMTs. Verify that the names in the RDMT name field of the LOGSAVE record are typed correctly and that they are in the RFT data set allocated to your session-parameters member.</td>
</tr>
<tr>
<td>20</td>
<td>56</td>
<td>Error freeing the RFT data set.</td>
</tr>
</tbody>
</table>
Table 40 (Page 4 of 6). BLMSSGEN Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>4</td>
<td>A database error occurred reading the LOGSAVE record.</td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>The LOGSAVE record was not found.</td>
</tr>
<tr>
<td>24</td>
<td>12</td>
<td>An enqueue error occurred reading the LOGSAVE record.</td>
</tr>
<tr>
<td>24</td>
<td>16</td>
<td>The database contains more than one LOGSAVE record.</td>
</tr>
<tr>
<td>24</td>
<td>20</td>
<td>Not enough storage to read the LOGSAVE record.</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>The LOGSAVE record is busy.</td>
</tr>
<tr>
<td>24</td>
<td>28</td>
<td>Cannot read the LOGSAVE record. The record is damaged. Delete the record and recreate it.</td>
</tr>
<tr>
<td>24</td>
<td>32</td>
<td>Record with RNID LOGSAVE is not LOGSAVE record type. Delete it and create the LOGSAVE record using the procedures described in Propagating to an Information/Management Database and Propagating to a DB2 Database.</td>
</tr>
<tr>
<td>24</td>
<td>36</td>
<td>No RDMTs specified in the Mapping field in the LOGSAVE record. BLMSSGEN stops at the first empty slot in the list of RDMTs. Verify that the first slot is not empty.</td>
</tr>
<tr>
<td>24</td>
<td>40</td>
<td>The send Data set name field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>24</td>
<td>44</td>
<td>The SQL data set name field was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>Return Code (TSCAFRET)</td>
<td>Reason Code (TSCAFRES)</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>24</td>
<td>48</td>
<td>The <strong>Logical record length</strong> field for the SQL data set was not found in the LOGSAVE record, or the length is 0. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>24</td>
<td>52</td>
<td>The <strong>Block size</strong> field for the SQL data set was not found in the LOGSAVE record, or the size is 0. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>24</td>
<td>56</td>
<td>The <strong>Primary allocation blocks</strong> field for the SQL data set was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>24</td>
<td>64</td>
<td>The <strong>Unit type</strong> field for the SQL data set was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>24</td>
<td>68</td>
<td>The <strong>ABEND disposition</strong> field for the SQL data set was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>24</td>
<td>72</td>
<td>Cannot read the LOGSAVE record. Unknown reason.</td>
</tr>
<tr>
<td>24</td>
<td>76</td>
<td>The <strong>Authorization ID</strong> field for the DB2 table was not found in the LOGSAVE record. Either the information is not in the LOGSAVE record, or the TSCA variable data area is damaged.</td>
</tr>
<tr>
<td>28</td>
<td>8</td>
<td>Error allocating the send data set. Look for other messages.</td>
</tr>
<tr>
<td>28</td>
<td>12</td>
<td>Error opening the send data set. Look for other messages.</td>
</tr>
<tr>
<td>28</td>
<td>16</td>
<td>Error closing the send data set. Look for other messages.</td>
</tr>
</tbody>
</table>
Table 40 (Page 6 of 6). BLMSSGEN Return and Reason Codes

<table>
<thead>
<tr>
<th>Return Code (TSCAFRET)</th>
<th>Reason Code (TSCAFRES)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>28</td>
<td>28</td>
<td>Error reading the send data set. Look for other messages.</td>
</tr>
<tr>
<td>28</td>
<td>56</td>
<td>Error freeing the send data set.</td>
</tr>
<tr>
<td>32</td>
<td>8</td>
<td>Error allocating the SQL data set. Look for other messages.</td>
</tr>
<tr>
<td>32</td>
<td>12</td>
<td>Error opening the SQL data set. Look for other messages.</td>
</tr>
<tr>
<td>32</td>
<td>16</td>
<td>Error closing the SQL data set. Look for other messages.</td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>Error writing the SQL data set. Look for other messages.</td>
</tr>
<tr>
<td>32</td>
<td>56</td>
<td>Error freeing the SQL data set.</td>
</tr>
<tr>
<td>40</td>
<td>4</td>
<td>Cannot locate database 5. Verify that you specified the correct session-parameters member.</td>
</tr>
</tbody>
</table>
The Information/Management Archiver

The Archiver is an Information/Management utility which allows you to manage records on your Information/Management databases. The following functions are provided:

Archive: Move records from one database to another. Refer to “Sample 1” on page 15-18 for sample BLGARCP specifications.

Copy: Copy records from one database to another (refer to “Sample 2” on page 15-19) or copy records from one database back to the same database or to another database with a new RNID (refer to “Sample 3” on page 15-20).

Purge: Delete records from a database. Refer to “Sample 4” on page 15-21 for sample BLGARCP specifications.

Note: The term “live” database refers to the existing, or “from” database, and the “archive” database refers to the new, or “to” database.

The advantage of using the Archiver instead of other methods of manipulating records on a database is that you can run the Archiver against a database while other users are interactively using Information/Management on the same database. You are not required to run BLGUT1 or any other Information/Management utilities in conjunction with running the Archiver.

In addition, 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 has successfully done
- Full LLAPI (Low-Level Application Programming Interface) logging
- Internal analysis of each record processed with details of all errors encountered

Functional Overview

The Archiver is a LLAPI job that makes use of dynamic program interface data tables (PIDTs). You can run the Archiver while other users are interactively accessing the databases. The “archive” database must be accessed as database 5, read/write. By default, the “live” database is also accessed as database 5,
read/write, but it may be accessed as read-only if the **DBID=** parameter is specified during **FUNCTION=COPY**.

The Archiver’s functions are controlled by parameters passed to it at invocation. The sample JCL shown in Figure 22 should be modified to fit your environment. Once modified, it can be used to invoke the Archiver.

```plaintext
//BLGARC JOB
//*/--------------------------------------------------------------------------*/
//*/ Licensed Materials - Property of Tivoli Systems */
//*/ 5648-142 (C) Copyright Tivoli Systems Inc., an IBM Company 1997 */
//*/ See Copyright Instructions */
//*/--------------------------------------------------------------------------*/
//*/ This is a sample jobstream for running the archiver, BLGARC. */
//*/--------------------------------------------------------------------------*/
//ARCHIVE EXEC PGM=BLGARC,REGION=6M
//*/ Do not forget to STEPLIB to your session load library if needed.
//*/ STEPLIB DD DSN=BLM.V1R1M/zerodot.SBLMMOD1,DISP=SHR
//*/ BLGARCP DD is the input parameters data set and is required.
//*/ BLGARCP DD DSN=?????????.BLGARCP,DISP=SHR
//*/ BLGINREC DD is needed only if INPUT=EXTERNAL is coded in the
//*/ data set defined by the BLGARCP DD.
//*/ BLGINREC DD DSN=?????????.BLGINREC,DISP=SHR
//*/ BLGOPREC DD is the data set where the RNIDs of the
//*/ records archived (created) are stored.
//*/ This data set can be used an input to another archive job.
//*/ BLGOPREC DD DSN=?????????.BLGOPREC,DISP=OLD
//*/ RFTDD DD HAS YOUR PIDT EXECUTABLE TABLES
//*/ RFTDD DD DSN=BLM.V1R1M/zerodot.SBLMFMT,DISP=SHR
//*/ APIPRINT DD SYSOUT=/
//*/ BLGOUT DD is the data set where Archiver messages are routed.
//*/ BLGOUT DD SYSOUT=/
//*/ SYSDUMP DD SYSOUT=/
//*/ SYSPRINT DD SYSOUT=/
//*/--------------------------------------------------------------------------*/
```

Figure 22. Sample JCL to Invoke the Archiver. The sample JCL can be found in the SBLMSAMP data set as member BLGARCJ.

The parameters enable you to initialize and control the use of the Program Interface Communications Area (PICA), which in turn controls the functions available within the LLAPI. For example, the parameters you specify in the **BLGARCP** data set define such items as the number of records to process and the search argument to be used for inquiry, as well as events such as the transfer of text and journal data in the records. (All of the parameters and their values are described in “Input Parameters” on page 15-6.)
The Archiver automatically updates both the SDDS (structured description data set) and SDIDS (structured description index data set) so there is no need to run any additional utilities, such as BLGUT1.

The Archiver uses several PIDTs to control inquiry, retrieve, and create functions. The sample BLGARCP shown in Figure 24 on page 15-18 uses the inquiry PIDT for problem records, BLGYPRI, which is a member of the SBLMFMT data set. The source code for this data table (PIDT) and other PIDTs can be found in the SBLMSAMP library. The SBLMFMT and SBLMSAMP libraries are described in the Planning and Installation Guide and Reference. You should review the sample PIDTs to ensure that they contain the values that you need. If they do not, you should create your own, modeled on the samples. More information on modifying PIDTs can be found in the Application Program Interface Guide.

The inquiry PIDT can contain a high level inquiry only, such as the record type. The specific search argument can be passed to the Archiver as a parameter list to form a freeform search. The retrieve and create PIDTs are built dynamically by the LLAPI, and automatically represent an exact data map of the retrieved record. Once retrieved, each record can be compressed and entries removed or unrecognized prior to the create transaction. In addition, the create transaction can be requested to always issue a new system-assigned RNID.

To determine the records to be processed, the Archiver can use either specific search arguments or an external data set.

- Using search arguments, an inquiry is performed using the inquiry PIDT in conjunction with the freeform argument specified by the ARG= parameters in the parameter data set BLGARCP. The values supplied by the ARG= parameters form the freeform search argument for the inquiry. An inquiry PIDT must contain the s-word (structured word) of the record type to be processed.

- The external data set utilizes a “header” line and explicit record identifiers. The data set contains a sequential list of Information/Management record number identifiers (RNIDs). This data set can be created “manually” or it can be created with an RFT (report format table) which could perform all the necessary logic searching, especially when processing complex parent/child relationships. Refer to Figure 35 on page 15-27 for sample RFT code to build the BLGINREC data set.

The Archiver logs the startup parameters that you specified, messages related to each record processed, and a completion summary. Logging is controlled by the LOGGING= parameter specified in the BLGARCP parameter data set. Archiver messages are written to the data set defined by the BLGOUT DD statement and LLAPI messages are written to the data set defined by the APIPRINT DD statement, assuming that these data sets were allocated and LOGGING=Y was specified in the BLGARCP parameter data set.

The Archiver creates the requested records in the “archive” database and automatically checks them in on both databases when CHECKIO=Y.
Starting in Version 1.1, the Archiver includes the following internal improvements:

- The Archiver manages non-zero PIDTCODEs, PIHTCODEs, and PIRT CODEs by analyzing each record as it is retrieved. A message such as BLG01127I, BLG01128I, or BLG01131I is issued to indicate the specific type of error encountered. The Archiver skips the record in error and continues until all records have been processed.

- When errors are encountered in the BLGARCP data set, message BLG01130E is issued stating the parameter in error. In earlier versions of Information/Management, a 701 abend was issued.

- The Archiver issues a message identifying the LLAPI environment which caused the Archiver to terminate (LIVE|ARCHIVE).

The Archiver also includes the following new parameters to provide the described functional enhancements:

- **DBID** Defines a read-only database (4,7,8, or 9) for the LIVESESS to enable you to copy records from your read-only database to your read/write database.

- **ARCCLASS** Defines the Privilege Class name that exists on the archive database.

- **DELENTRY** Removes redundant, not valid, or unwanted entries from records.

- **UNCOGNIZE** Defines whether to uncognize entries in a record.

- **COMPRESS** Removes non-essential information from records.

- **NEWRNID** Creates multiple copies of the same record on the same database or on another database.

- **USERRELN** Defines whether you wish to process records with user-defined relationships.

- **TIMEOUT** Defines a time value during which the transaction must complete.

Refer to “Input Parameters” on page 15-6 for additional information on these parameters.

**Note:** The API does not support multiple response list processor data. By multiple response list processor data, we mean data collected by the list processor which is prefixed and contains more than one response per file. Records that contain multiple response list processor data cannot be processed by the Archiver because data loss or data corruption may occur. String data is not considered multiple response list processor data.
Input Data Sets

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGARCP</td>
<td>Input parameter data set; it may be specified in-stream if desired. See the example in Figure 24 on page 15-18. It should have the following characteristics: RECFM=FB LRECL=80</td>
</tr>
<tr>
<td>BLGINREC</td>
<td>Input data set containing the RNIDs to be processed when INPUT=EXTERNAL is specified. The first line in the data set, beginning in column 1, should contain the word ARCHIVE or COPY or PURGE matching the FUNCTION= value in the BLGARCP DD statement. The input may be specified in-stream if desired. It should have the following characteristics: RECFM=FB LRECL=80</td>
</tr>
<tr>
<td>RFTDD</td>
<td>Input data set containing the PIDTs.</td>
</tr>
</tbody>
</table>

Output Data Sets

<table>
<thead>
<tr>
<th>DDNAME</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGOPREC</td>
<td>Output data set containing the RNIDs copied as part of user-defined relationship processing. The first record written to the BLGOPREC data set is always a PURGE entry. This data set should be used as the BLGINREC data set for a subsequent PURGE. You should allocate the BLGOPREC data set prior to running the Archiver, and it should have the following characteristics: RECFM=FB LRECL=80</td>
</tr>
<tr>
<td>BLGOUT</td>
<td>Output data set where all Archiver messages are written. It should have the following characteristics: RECFM=FB LRECL=80 BLKSIZE=80</td>
</tr>
<tr>
<td>APIPRINT</td>
<td>Output data set where all LLAPI messages are written when LOGGING=Y is specified. The Application Program Interface Guide contains additional information on the APIPRINT data set.</td>
</tr>
</tbody>
</table>

Return Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The Archiver processed all of the requested records successfully.</td>
</tr>
<tr>
<td>4</td>
<td>The Archiver encountered a non-zero PIDTCODE, PIHTCODE, or PIRTCODE. The record in error is skipped, and processing continues until all of the requested records have been processed. Archiver messages are issued describing the exact failure.</td>
</tr>
</tbody>
</table>
8 The Archiver encountered an error and cannot continue. Archiver messages are issued describing the exact failure.

Input Parameters

Attention: If you specify any of these parameters more than once, only the last value specified is be used by the Archiver program. In certain circumstances, this can also cause unpredictable results.

These are the parameters that can be specified with the Archiver:

FUNCTION= Defines the function of the Archiver program. Valid options are:

ARCHIVE Move the specified records from the live database to the archive database and delete the original record from the live database.

COPY Copy the specified records from the live database to the archive database. The records on the live database are left intact.

PURGE Delete the specified records from the live database. The archive database is not required or affected. If the Archiver encounters a record that cannot be found (Return Code 12 Reason Code 10 on the API delete transaction), the record is skipped, the Archiver continues processing with the next record, and the Archiver finishes with return code 4 (RC4).

The default is ARCHIVE.

PURGE= Defines the type of delete processing to be used by the ARCHIVE option. Valid values are:

NORMAL Delete the record from the live database only if the record is created on the archive database with a Return Code=0.

FORCE Delete the record from the live database if the record is created on the archive database with a Return Code=0 OR if the create function fails because a record already exists on the archive database with the same RNID.

The default is NORMAL.

INPUT= Defines the source of the RNIDs to be processed. Valid values are:

INTERNAL Use the search arguments specified in the input parameter member (up to 10 ARG= may be specified) in association with the INQPIDT parameter to locate the required records on the live database.
EXTERNAL Use the RNIDs specified in the input data set allocated to DDNAME BLGINREC. The first record read from DDNAME BLGINREC must match the function specified in the FUNCTION= parameter.

The default is INTERNAL.

CLASS= Defines one Privilege Class name that exists on both live and archive databases and is used for live and archive sessions. This class must have Inquiry, Create, Update, Delete, and database administration authority to the specified records on the live database, and Create, Update, and database administration authority on the archive database.

You can override the Privilege Class name on the archive database with the ARCCLASS parameter.

This is a Required parameter, and no default value is supplied.

Sets PICAPRIV for the API transactions that will be performed on the archive database. Refer to the Applications Program Interface Guide and Reference for additional information on the PICAPRIV field.

ARCCLASS= Defines the Privilege Class name that exists on the archive database. This privilege class must have Create, Update, and database administration authority to the specified records on the archive database.

This is an Optional parameter, and no default value is supplied.

Sets PICAPRIV for the API transactions that will be performed on the archive database. Refer to the Applications Program Interface Guide and Reference for additional information on the PICAPRIV field.

LIVESESS= Defines the session parameter member name to access the live database; for example, BLGSES00. The “live” database must be accessed as database 5, the read/write database, if FUNCTION=ARCHIVE or FUNCTION=PURGE. With FUNCTION=COPY, the database can be accessed as read-only if desired assuming that CHECKIO=Y is not specified.

This is a Required parameter, and no default value is supplied.

ARCSESS= Defines the session parameter member name to access the archive database; for example, BLGSESAR. The “archive” database must be accessed as database 5, the read/write database. The archive session is only initialized with FUNCTION=ARCHIVE and FUNCTION=COPY.

This is a Required parameter if FUNCTION=ARCHIVE or FUNCTION=COPY, and no default value is supplied.
DBID=
Defines a read-only database (4, 7, 8, or 9) for the LIVESESS. Specifying this parameter enables you to copy records from your read-only database to your read/write database. To do this, you use the same session parameters member for both the LIVESESS and the ARCSESS. This is only valid when FUNCTION=COPY.

This parameter is Optional and is ignored if FUNCTION=COPY is not specified. Database 5 is assumed for both sessions if DBID is not specified. This parameter sets PICADBID for the API transactions that will be performed on the live database. Refer to the Applications Program Interface Guide and Reference for additional information on the PICADBID field.

INQPIDT=
Defines the name of the inquiry PIDT used to locate the records to be processed when INPUT=INTERNAL. This PIDT is used in association with the search arguments specified in the ARG= parameters.

This is a Required parameter when INPUT=INTERNAL, and this parameter is Ignored when INPUT=EXTERNAL. No default value is supplied.

DYNMPIDT=
Defines the name of the dynamic PIDT used by the Retrieve and Create functions to copy and load the record into the archive database. This PIDT should only contain the USE(HEADER) parameter. The PIDT BLGYDYN is supplied in SBLMFMT for this purpose and the default is DYNMPIDT=BLGYDYN.

This is a Required parameter.

LOGGING=
Defines whether to perform API logging. Valid values are:

Y  A value of Y causes full logging to DDNAME APIPRINT.

N  A value of N suppresses logging.

The default is N. This parameter sets the PICAMSGD parameter to P. For additional information, refer to the Application Program Interface Guide.

ARG=
Defines up to 10 freeform arguments that are used when INPUT=INTERNAL. The maximum length of each argument (including the qualifier) is 33 characters. The arguments specified are used in addition to the inquiry PIDT. Each argument must be prefixed by an ARG= statement. A period (.) at the end of an argument specifies an abbreviated search. A Boolean operator, if specified, must be in the first position of the argument. An argument cannot contain leading or imbedded blanks. Examples of arguments are:
Information/Management Archiver

Input Parameters

ARG=CLOSED
ARG=DATE/94/01/01
ARG=DATE/94/12/31
ARG=PRIO/1
ARG=PRIO/1
ARG=-DEPT/T53B
ARG=STAC/CL.
ARG=DATE/95/**/**

This parameter is Optional and it is ignored when INPUT=EXTERNAL. No default value is supplied. Although this parameter is optional, you may want to specify it to more closely identify the data you want to process. If you omit this parameter, the Archiver searches against all of the records defined by the INQPIDT= parameter.

APPLID=
Defines the application identifier (APPLID) of the Archiver, and must exist in the Privilege Class defined by the CLASS parameter and the ARCCCLASS parameter (if specified). This parameter is Optional, and the default is APPLID=ARCHIVER.

CHILD=
Defines the p-word (prefix word), including the slash (/), that is to be used to locate child records by performing a search of p-word/parent_rnid. The p-word must be 5 characters in length, including the slash. Omitting this parameter causes child processing to be bypassed. When using INPUT=INTERNAL and specifying this parameter, each record is checked for associated child records. This can seriously affect the performance of the Archiver. This method should only be used for IBM-defined parent/child relationships. See “Parent/Child Relationships” on page 15-23 for additional information. Multiple child types are located with the CHILD= argument. If you have multiple child record types associated with a single parent, you should code an RFT to locate all of the child RNIDs.

This parameter is Optional, but if you specify this parameter, you must also specify the PARENT= and CHDPIDT= parameters.

PARENT=
Defines the s-word (structured word) index pointer contained in the parent record pointing to the child record. Usually the associated p-word is NAMA. This enables the inquiry function to load the pointer to the child record without having to perform a search against every parent. If associated data is located, a search is performed and all resulting child records are also processed. This method should only be used for IBM-defined parent/child relationships. See “Parent/Child Relationships” on page 15-23 for additional information.

This parameter is Optional, but if you specify this parameter, you must also specify the CHILD= and CHDPIDT= parameters.
**CHDPIDT=** Defines the name of the PIDT used to locate child records. Usually, this PIDT would contain the child record type s-word. This method should only be used for IBM-defined parent/child relationships. See “Parent/Child Relationships” on page 15-23 for additional information.

This parameter is **Optional**, but if you specify this parameter, you must also specify the **PARENT=** and **CHILD=** parameters.

**USERRELN=** Defines whether or not you wish to process records with user-defined relationships. The valid values are:

- **Y** The records to be processed contain user-defined relationships. Allocate the **DDNAME BLGOPREC** in the **BLGARC** JCL. As each record is processed, its RNID is written to the **BLGOPREC** data set.

- **N** The records to be processed do not contain user-defined relationships.

This is an **Optional** parameter, but if you specify **Y** then **FUNCTION=COPY** and **INPUT=EXTERNAL** must also be specified. The default is **N**. You should specify **CHECKIO=Y** when using this parameter.

See “Parent/Child Relationships” on page 15-23 for additional information.

**HITS=** Defines the number of parent records to be processed. The value specified must be an 8-digit decimal number between 00000001 and 99999999. Child records are not included in this number. This parameter controls execution times by limiting the number of records to be processed.

This parameter is **Ignored** when **INPUT=EXTERNAL** and defaults to 00000001 when **INPUT=INTERNAL**. The PICA field **PICANUMH** is set by the **HITS=** parameter.

**CHECKIO=** Defines whether to perform check-in/check-out for each record processed. This parameter allows you to check out records on the live database before starting the archive process. The valid values are:

- **Y** Specifies that records should be checked out before archive processing starts and checked in when the archive process completes. The API check-in transaction is not performed to check in the records created on the archive database. Instead, the checked out entry in a retrieved record is ignored on the subsequent create transaction.

- **N** Specifies that no check-in or check-out processing is requested.
This parameter is **Optional**, and the default is **Y**.

If you do not specify **CHECKIO=N**, the default of **CHECKIO=Y** is taken and may cause unnecessary overhead.

Setting **CHECKIO=Y** prevents two integrity problems:

- It prevents a user from updating a record on the live database between the retrieve and delete transactions.
- It prevents the delete transaction from failing because a user is updating the record.

It is unlikely that either of these situations will occur during normal archive processing.

When using **FUNCTION=COPY** or **FUNCTION=PURGE**, setting this parameter to **N** can greatly increase the throughput of records processed, but setting this parameter to **N** with **FUNCTION=ARCHIVE** may cause data integrity problems. You should always specify **CHECKIO=Y** when **USERRELN=Y**. See “Parent/Child Relationships” on page 15-23 for additional information.

**COMPRESS**= Defines whether to remove non-essential information from records.

Valid values are:

- **Y** Compress entries from the records.
- **N** Do not compress entries from the records.

This parameter is **Optional**, and the default value is **N**.

Entries in the records which contain a **PIDTRDEF** value of **O** are removed before the archive version of the record is created. These entries signify that the SDE for the row does not contain an s-word or a p-word. Refer to the *Application Program Interface Guide* for additional information on **PIDTRDEF**.

Refer to “Record Processing Considerations” on page 15-29 for additional information on when to code this parameter.

**DELENTRY**= Defines whether to remove redundant, not valid, or unwanted entries from records. Defines up to 40 s-word indexes which identify the data in your records that you want deleted from your records before they are created on the archive database. For example, if you want to delete the reporter name and status from a problem record, specify:

```
DELENTRY=S/zerodotB59
DELENTRY=S/zerodotBEE
```

The s-word indexes that you specify are compared with the **PIDTSYMB** field of each **PIDT** entry for each record. An s-word index match causes the **PIDT** entry to be ignored when the record is created on the archive database. The s-word index specified must
be 5 characters prefixed with an S. An s-word index of S0000 is ignored.

This parameter is **Optional**, and no default value is supplied.

Refer to “Record Processing Considerations” on page 15-29 for additional information on when to code this parameter.

**UNCOGNIZE=** Defines whether to uncognize entries in a record. Defines up to 40 s-word indexes which identify the data that you want uncognized in your records before they are created on the archive database. For example, if you want to uncognize the reporter name and status from a problem record, specify:

```
UNCOGNIZE=S/zerodotB59
UNCOGNIZE=S/zerodotBEE
```

The s-word indexes that you specify are compared with the PIDTSYMB field of each PIDT entry for each record. An s-word index match causes the PIDT entry to be uncognized when the record is created on the archive database. The s-word index specified must be 5 characters prefixed with an S. An s-word index of S0000 is ignored.

This parameter is **Optional**, and no default value is supplied.

Refer to “Record Processing Considerations” on page 15-29 for additional information on when to code this parameter.

**NEWRNID=** Defines whether the RNID of the record to be processed should be replaced with the next available system-assigned RNID on the archive database. This is valid for `FUNCTION=_COPY` and `FUNCTION=ARCHIVE`.

**Y** The next system-assigned RNID on the archive database replaces the RNID currently assigned to the record when it is created on the archive database.

**N** Keep the RNID of the record.

This parameter is **Optional** and the default value is **N**.

**Note:** By specifying `NEWRNID=Y`, the RNID of the retrieved record is always replaced by the next system-assigned RNID on the archive database regardless of whether or not the RNID exists on the archive database or if the record has a user-assigned RNID. Do not set `NEWRNID=Y` for records which contain relationships, or the relationships will be broken.

**TEXT=** Defines whether freeform text from the records being processed should be copied. Valid values are:

**Y** Process the freeform text with the record.

**N** Do not process the freeform text with the record.
This parameter is **Optional** and the default is **Y**.

**TEXTXFER=** Defines the transfer method used when processing record freeform text. Valid values are:

- **D** Use a data set to transfer the freeform text.
- **B** Use a buffer to transfer the freeform text.

This parameter is **Optional** and the default is **D**.

For performance reasons, a value of **B** is desirable. If you specify **D**, it is your responsibility to delete the data sets created by the archive job. For more details, see **PICATXTP** in the *Application Program Interface Guide*.

**TEXTAUD=** Defines whether the freeform text audit trail should be processed when processing record text. Valid values are:

- **Y** Preserve the freeform text audit trail.
- **N** Discard the freeform text audit trail.

This parameter is **Optional** and the default is **Y**.

For additional details, see **PICATXAU** in the *Application Program Interface Guide*.

**TEXTCNT=** Defines the number of text lines to process when **TEXTXFER=B**.

This value must be 8 characters in length, containing leading zeroes as necessary. The default is **00000200**.

For additional details, see **PICATXTU** in the *Application Program Interface Guide*.

**TEXTWID=** Defines the width of the text to process when **TEXTXFER=B**.

This value must be 8 characters in length, containing leading zeroes as necessary. The default is **00000060**.

For additional details, see **PICATXTW** in *Application Program Interface Guide*.

**TEXTAREA=** Defines the area of text to transfer when using **TEXTXFER=B** and the total number of text lines exceeds the amount specified in **TEXTCNT**. Valid values are:

- **T** The top of the text is copied.
- **B** The bottom of the text is copied.

The default is **B**.

For additional details, see **PICATXTA** in the *Application Program Interface Guide*.
**TIMEOUT=** Defines a value which is the number of seconds in which the transaction must complete. If you specify a value of 0 or omit this field, the value is set to 00000300 seconds (five minutes). This value sets the **PICATINT** parameter, which is defined in the *Application Program Interface Guide*.

**HISTORY=** Defines whether to process history (journal) data in the record. Valid values are:

- **Y** Preserve the history data.
- **N** Discard the history data.

The default is **Y**.

For additional details, see **PICAHIST** in the *Application Program Interface Guide*.
# Required Parameters

Depending on the FUNCTION= specified, certain parameters may be required.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>FUNCTION=</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARCHIVE</td>
<td>COPY</td>
<td>PURGE</td>
</tr>
<tr>
<td>Required</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>PURGE=</td>
<td>Optional</td>
<td>N/A</td>
</tr>
<tr>
<td>INPUT=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>CLASS=</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>ARCCCLASS=</td>
<td>Optional</td>
<td>N/A</td>
</tr>
<tr>
<td>LIVESESS=</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>ARCCSESS=</td>
<td>Required</td>
<td>N/A</td>
</tr>
<tr>
<td>APPLID=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>INQPIDT=</td>
<td>Required</td>
<td>Required</td>
</tr>
<tr>
<td>INQPIDT=</td>
<td>Required if INPUT= INTERNAL</td>
<td>None</td>
</tr>
<tr>
<td>DBID=</td>
<td>N/A</td>
<td>Optional</td>
</tr>
<tr>
<td>DBID=</td>
<td>N/A</td>
<td>Optional</td>
</tr>
<tr>
<td>DYNMPIDT=</td>
<td>Required</td>
<td>N/A</td>
</tr>
<tr>
<td>LOGGING=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>ARG=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>CHILD=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>PARENT=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>CHDNPIDT=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>USERRELN=</td>
<td>N/A</td>
<td>Optional</td>
</tr>
<tr>
<td>HITS=</td>
<td>Required if INPUT= INTERNAL</td>
<td>Required if INPUT= INTERNAL</td>
</tr>
<tr>
<td>CHECKIO=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>COMPRESS=</td>
<td>Optional</td>
<td>N/A</td>
</tr>
<tr>
<td>DELENTRY=</td>
<td>Optional</td>
<td>N/A</td>
</tr>
<tr>
<td>UNCOGNIZE=</td>
<td>Optional</td>
<td>N/A</td>
</tr>
<tr>
<td>NEWRNID=</td>
<td>Optional</td>
<td>N/A</td>
</tr>
<tr>
<td>TEXT=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>TEXTXFER=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
<tr>
<td>TEXTAUD=</td>
<td>Optional</td>
<td>Optional</td>
</tr>
</tbody>
</table>
### Required Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Optionality</th>
<th></th>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXTCNT=</td>
<td>Optional</td>
<td>Optional</td>
<td>Optional</td>
<td>00000200</td>
</tr>
<tr>
<td>TEXTWID=</td>
<td>Optional</td>
<td>Optional</td>
<td>Optional</td>
<td>00000060</td>
</tr>
<tr>
<td>TEXTAREA=</td>
<td>Optional</td>
<td>Optional</td>
<td>Optional</td>
<td>B</td>
</tr>
<tr>
<td>TIMEOUT=</td>
<td>Optional</td>
<td>Optional</td>
<td>Optional</td>
<td>00000300</td>
</tr>
<tr>
<td>HISTORY=</td>
<td>Optional</td>
<td>Optional</td>
<td>Optional</td>
<td>Y</td>
</tr>
</tbody>
</table>
**BLGINREC Sample**

The Archiver input data set expects the following conditions:

- The first line beginning in column 1 should contain the word *ARCHIVE* or *COPY* or *PURGE* matching the **FUNCTION** value in the data set named in the **BLGARCP DD** statement.
- Each RNID should begin in column 1 with only one RNID per line.
- All numeric RNIDs should have leading zeroes.

This data set is read and processed sequentially.

This sample **BLGINREC** could be used when **INPUT=EXTERNAL** in **BLGARCP DD** to archive only records 00000001, 00000002, and 00000003 (regardless of record type) into the *archive* database. Once these records are archived, they are deleted from the *live* database.

<table>
<thead>
<tr>
<th>ARCHIVE</th>
<th>* (Must be ARCHIVE or COPY or PURGE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000001</td>
<td>* (Only one record per line)</td>
</tr>
<tr>
<td>00000002</td>
<td>* (Numeric records must have leading zeroes)</td>
</tr>
<tr>
<td>00000003</td>
<td>*</td>
</tr>
</tbody>
</table>

Figure 23. Sample BLGINREC
BLGARCP Samples

The Archiver parameter data set expects the following conditions:

- Archiver parameter specifications begin in column 1
- Comments, if specified, begin in column 40
- Each line contains a parameter; blank lines are not allowed

This data set is read and processed sequentially.

Sample 1

This sample BLGARCP archives up to 1000 Problem records with a status of CLOSED that were entered in 1995 from the “live” database defined in session BLGSES00 to the “archive” database defined in session BLGSESAR. In this example, the records are copied and compressed to the database defined in BLGSESAR and deleted from the database defined in BLGSES00.

<table>
<thead>
<tr>
<th>FUNCTION=ARCHIVE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PURGE=NORMAL</td>
</tr>
<tr>
<td>INPUT=INTERNAL</td>
</tr>
<tr>
<td>CLASS=SYSADMIN</td>
</tr>
<tr>
<td>LIVESESS=BLGSES00</td>
</tr>
<tr>
<td>ARCSCESS=BLGSESAR</td>
</tr>
<tr>
<td>APPLID=ARCHIVER</td>
</tr>
<tr>
<td>INQPIDT=BLGYPRI</td>
</tr>
<tr>
<td>DYNMPIDT=BLGYDYN</td>
</tr>
<tr>
<td>LOGGING=Y</td>
</tr>
<tr>
<td>ARG=STAC/CLOSED</td>
</tr>
<tr>
<td>ARG=DATE/97/01/01</td>
</tr>
<tr>
<td>ARG=DATE/97/12/31</td>
</tr>
<tr>
<td>HITS=00001000</td>
</tr>
<tr>
<td>CHECKIO=N</td>
</tr>
<tr>
<td>COMPRESS=Y</td>
</tr>
<tr>
<td>TEXT=Y</td>
</tr>
<tr>
<td>TEXTAUD=Y</td>
</tr>
<tr>
<td>TEXTXFER=B</td>
</tr>
<tr>
<td>TEXTCNT=00001000</td>
</tr>
<tr>
<td>TEXTWID=00000132</td>
</tr>
<tr>
<td>TEXTAREA=B</td>
</tr>
<tr>
<td>HISTORY=Y</td>
</tr>
</tbody>
</table>

Figure 24. Sample BLGARCP
Sample 2

This sample **BLGARCP** copies the records with RNID/00063542 and RNID/00064967 from the database defined as database 4 in session **BLGSES00** to the database defined as database 5 in session **BLGSES00**. It is assumed that RNID/0003542 and RNID/00064967 do not exist in database 5. Before the records are created on database 5, the records are compressed, and the Date and Time Occurred data values are removed from the records.

```plaintext
FUNCTION=COPY
INPUT=INTERNAL
CLASS=SYSADMIN
LIVESESS=BLGSES00
ARCSSESS=BLGSES00
DBID=4 /* Records read from DB 4
APPLID=ARCHIVER
INQPIDT=BLGYPRI
DYNMPIDT=BLGYDYN
LOGGING=Y
ARG=RNID/00063542 /* Search for two specific records
ARG=RNID/00064967
HITS=/zerodot/zerodot/zerodot/zerodot1/zerodot/zerodot/zerodot
CHECKIO=N /* Default is Y
COMPRESS=Y /* Default is N
DELENTRY=S/zerodotC3D /* Delete Date Occurred
DELENTRY=S/zerodotC6A /* Delete Time Occurred
TEXT=Y
TEXTAUD=Y
TEXTXFER=B /* Default is D
TEXTCNT=00001000 /* Default is 200
TEXTWID=00000132 /* Default is 60
TEXTAREA=B
HISTORY=Y
```

Figure 25. Sample BLGARCP
Sample 3

This sample **BLGARCP** copies the record with `RNID/00051368` from the database defined as database 5 in session **BLGSES00** to the database defined as database 5 in session **BLGSES00** (that is, the same database). When the record is created on database 5, it will be compressed and assigned the next available system-assigned RNID.

```plaintext
FUNCTION=COPY
INPUT=INTERNAL
CLASS=SYSADMIN
LIVESESS=BLGSES00      /* Read from and write to DBS5
ARCSESS=BLGSES00
APPLID=ARCHIVER
INQPIDT=BLGPRI
DYNMPIDT=BLGYPDN
LOGGING=Y
ARG=RNID/00051368      /* Copy 1 record
HITS=00001000
CHECKIO=N               /* Default is Y
COMPRESS=Y              /* Default is N
NEWRNID=Y               /* RNID of record is replaced
TEXT=Y
TEXTAUD=Y
TEXTXFER=B              /* Default is D
TEXTCNT=00001000        /* Default is 200
TEXTWID=00000132        /* Default is 60
TEXTAREA=B
HISTORY=Y
```

Figure 26. Sample BLGARCP
Sample 4

This sample **BLGARCP** deletes the records specified in the **BLGINREC DD** in the **BLGARCJCL** from database 5 in session **BLGSES00**.

```
FUNCTION=PURGE
INPUT=EXTERNAL
CLASS=SYSADMIN
LIVESESS=BLGSES00
APPLID=ARCHIVER
LOGGING=N
CHECKIO=N
/* Default is Y
```

Figure 27. Sample BLGARCP

```
PURGE
00051368
00061854
00021537
00021336
```

Figure 28. Sample BLGINREC
Sample 5

This sample **BLGARCP** copies the record specified in the **BLGINREC DD** in the **BLGARC JCL** from database **4** in session **BLGSES00** to database **5** in session **BLGSES00** 8 times. Each time the record is created on database **5**, it is compressed and assigned the next available system-assigned RNID. Freeform text and history data are not copied.

```
FUNCTION=copy
INPUT=EXTERNAL /* BLGINREC has multiple entries
CLASS=SYSADMIN
LIVESESS=BLGSES00
ARCHSESS=BLGSES00
DBID=4
APPLID=ARCHIVER
DYNMPIDT=BLGYDYIN
LOGGING=Y
CHECKIO=N /* Default is Y
COMPRESS=Y /* Default is N
NEWRNID=Y /* RNID of record is replaced
TEXT=N
TEXTXFER=B
HISTORY=N
```

Figure 29. Sample BLGARCP

```
COPY
00051368
00051368
00051368
00051368
00051368
00051368
00051368
00051368
```

Figure 30. Sample BLGINREC

For samples of IBM-defined and User-defined relationship processing, refer to “Parent/Child Relationships” on page 15-23.
Parent/Child Relationships

IBM-defined Relationships

This sample BLGARCP copies the first 5000 Change records and their associated Activity records that were created in 1995 from database 5 in BLGSES00 to database 5 in BLGSESAR. After the records are successfully copied, they are deleted from database 5 in session BLGSES00. This technique only works for Change (parent) and Activity (child) records which were created with the IBM-supplied program exits or which adhere to the following rules:

- The Change records must contain an s-word with associated-data indicating that Activity records exist. For example, Change records created with the IBM-supplied program exits contain the s-word index X'0CBC' with p-word NAMA/ which point to a unique Activity name.

- The Activity records must contain a 5-character p-word (including the slash) with the RNID of the Change record. For example, Activity records created with the IBM-supplied program exits contain the p-word RNOR/.

```
FUNCTION=ARCHIVE
PURGE=NORMAL
INPUT=INTERNAL
CLASS=SYSADMIN
LIVESESS=BLGSES00
ARCSESS=BLGSESAR
APPLID=ARCHIVER
INQPIDT=BLGYCHI
DYNMPIDT=BLGYDYN
LOGGING=N
ARG=STAC/CLOSED
ARG=DATE/97/01/01
ARG=DATE/97/12/31
CHILD=RNOR/
PARENT=S/0CBC
CHDPIDT=BLGYACI
HITS=00005000
CHECKIO=N
TEXT=Y
TEXTAUD=Y
TEXTXFER=B
TEXTCNT=00001000
TEXTWID=00000132
TEXTAREA=B
HISTORY=Y
```

Figure 31. Sample BLGARCP
User-defined Relationships

The sample JCL shown in Figure 32 on page 15-25 shows you how to specify the BLGARCP parameters and the BLGINREC RNIDs in-stream. Records with user-defined relationships which are specified by the BLGINREC DD are copied from database 5 in session BLGSES00 to database 5 in BLGSESAR, and the BLGOPREC data set is created in the COPY step of the JCL. All of the RNIDs that are related to each other must be listed. That is, for every parent specified, all of the children must also be specified.

As the records are copied, the BLGOPREC data set is updated with the RNIDs of the records that were created successfully on the archive database. Then, in the DELETE step of the JCL, the BLGOPREC data set created in the COPY step is specified as the BLGINREC data set, and the RNIDs specified in the BLGINREC data set are deleted.

You must enable CHECKIO processing to prevent records from being updated between the COPY and the DELETE steps. Therefore, the records will remain checked out on database 5 in session BLGSES00 until they are deleted in the DELETE step.
Figure 32. Sample JCL to invoke the Archiver for User-Defined Relationships. Part 1 of 2
// The BLGINREC DD specifies the RNIDs of the records to be copied.

/*
 * BLGINREC DD *
 */
COPY
PARENT1
CHILD11
CHILD12
PARENT2
CHILD21
CHILD22
PARENT3
CHILD31
PARENT4
/*
/*
//DELETE EXEC PGM=BLGARC,REGION=4M
/*
/* Do not forget to SETPLIB to your session load library if needed.
/*
//STEPLIB DD DSN=BLM.V7R1M.SBLMMOD1,DISP=SHR
/*
/* This data set was created in the COPY step of this job by the
/* BLGOPREC DD.
/*
//BLGINREC DD DSN=?????????.BLGOPREC,DISP=OLD
/*
/* RFTDD DD has your PIDT executable tables
/*
//RFTDD DD DSN=BLM.V7R1M0.SBLMFMT,DISP=SHR
//APIPRINT DD SYSOUT=
//BLGOUT DD SYSOUT=
//SYSMDUMP DD SYSOUT=
//SYSPRINT DD SYSOUT=
//SYSOUT DD SYSOUT=
//BLGARCP DD *
FUNCTION=PURGE
INPUT=EXTERNAL
CLASS=SYSADMIN
LIVESESS=BLGSESAR
APPLID=ARCHIVER
LOGGING=N
/*

Figure 33. Sample JCL to invoke the Archiver for User-Defined Relationships. Part 2 of 2. The
sample JCL can be found in the SBLMSAMP data set as member BLGUSERJ.
Sample RFT

The RFT in Figure 35 works for both IBM-defined and user-defined relationships as long as the RNID of the Change (parent) record is contained in the Activity (child) records. For additional information on RFTs, refer to the Report Format Facility Guide and Reference.

```plaintext
SECTION
  PUT COL(001) VALUE(COPY)
  SEARCH ARGUMENT(!SO006, STAC/CLOSED DATE/96/01/01 - DATE/96/12/31) +
    MERGE(N)
    /* RNID of the change record */
    PUT COL(001) DATA(RNID/.)
  SET NAME(CHANGE) DATA(RNID/.)
    SEARCH ARGUMENT(!SO007 ~!SO006, RNOR/&CHANGE) MERGE(N)
    /* RNIDs of the activity records */
    PUT COL(001) DATA(RNID/.)
  ESEARCH
ESECTION
```

Figure 35. Sample RFT to produce the BLGINREC data set
Additional Tailoring Considerations

Performance Considerations

The following is a list of recommendations on how to improve the performance of the Archiver:

- Do not code STEPLIB DD statements in your JCL unless they are required. Place all Information/Management modules and session members in the Linklist or Linkpack Area. Refer to the Planning and Installation Guide and Reference for additional information. A STEPLIB DD statement causes all calls to load modules to access the STEPLIB DDNAME in an attempt to locate the modules, regardless of whether they reside in the data set(s).

- Disable CHECKIO processing unless USERRELN=Y or you specifically require records to be checked-out prior to retrieval. Coding CHECKIO=Y with FUNCTION=ARCHIVE or FUNCTION=COPY causes additional API transactions to be performed. When FUNCTION=COPY, each record will be checked out on the live database and then checked back in on the live database. When FUNCTION=ARCHIVE, each record will be checked out on the live database.

- When performing FUNCTION=ARCHIVE or FUNCTION=COPY, consider compressing the records by specifying COMPRESS=Y in the BLGARCP parameter data set. This removes redundant entries from each record and reduce the number of PIDT data entries to be processed by the API create transaction. In addition, DASD space is also saved on the archive database.

- When performing FUNCTION=ARCHIVE or FUNCTION=COPY, consider removing unwanted, not valid or redundant data entries by utilizing the DELENTRY parameter in the BLGARCP parameter data set. This reduces the number of PIDT entries to be processed by the API create transaction. In addition, if the entry deleted was cognized, updates to the SDIDS are also eliminated.

- When performing FUNCTION=ARCHIVE or FUNCTION=COPY, consider uncognizing fields that are no longer required as searchable by utilizing the UNCOGNIZE parameter in the BLGARCP parameter data set. This reduces the number of updates required to the SDIDS by the API create transactions.

- When performing FUNCTION=ARCHIVE or FUNCTION=COPY, specifying TEXTXFER=B eliminates the need for the LLAPI to allocate a data set for each text type in each record.

- If LLAPI logging is not required, specify LOGGING=N in the BLGARCP parameter data set. This eliminates the I/O to the APIPRINT data set caused during LLAPI logging.

- Consider defining separate archiver sessions with different characteristics to those used by the normal interactive users. Refer to the COGENQ parameter.
in the BLGCLUST macro and the SORTPFX parameter in the
BLGPARMS macro in the Planning and Installation Guide and Reference.

Record Processing Considerations

In order to use the DELENTRY, UNCOGNIZE, and COMPRESS parameters,
you must understand how the LLAPI processes the data in your records. When the
LLAPI retrieves a record from your live database, a dynamic program interface
data table (PIDT) is created. Each PIDT consists of a header and multiple data
rows which correspond to the data in your records. For the sake of simplicity, you
can use the output from the VIEW INTERNALS command to understand how
the PIDT data rows are loaded. Essentially, the lines of output from the VIEW
INTERNALS command are loaded line by line from the top to the bottom into
the PIDT data rows, and the Archiver assigns an entry number to each row which
corresponds to the VIEW INTERNALS line number. The SDDSROOT entry in
the VIEW INTERNALS output is not processed, and freeform text entries (1 per
freeform text type) are processed last. Refer to Figure 36 for a sample of the
output produced by the VIEW INTERNALS command.

<table>
<thead>
<tr>
<th>BLG1TVID VIEW INTERNAL DATA LINE 1 OF 30</th>
</tr>
</thead>
<tbody>
<tr>
<td>PANEL PANEL REL COG- FLAGS SWORD STRUCTURED PREFIX WORD OR</td>
</tr>
<tr>
<td>NAME TYP/RSP LEV NIZE F M D INDEX WORD VISIBLE PHRASE</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
</tr>
<tr>
<td>BLG0EN20 D/ 5 06 B/ 19/00/88 S002C BA //S/TS1 ENTRY</td>
</tr>
<tr>
<td>SDDSROOT C/ 00 N/ 12/04/10 S0000 0003E44</td>
</tr>
<tr>
<td>BLG00000 S/ 1 06 B/ 18/04/00 S0032 BA //S/TXS RECS=PROBLEM</td>
</tr>
<tr>
<td>BLG0B001 S/ 1 06 B/ 19/00/00 S0CFC BC IMĐIAENTO REPORTER</td>
</tr>
<tr>
<td>BLG0B100 D/ 1 06 N/ 41/00/00 S0B59</td>
</tr>
<tr>
<td>BLG6REQN A/ 2 06 B/U 00/04/00 S0B59 BC IM01OPS00 PERS/ANY/ONE</td>
</tr>
<tr>
<td>BLG0B100 D/25 06 N/ 41/00/00 S0E0F</td>
</tr>
<tr>
<td>BLG6DSAB A/ 2 06 B/U 00/04/00 S0E0F BC IM0TXCA00 TEST</td>
</tr>
<tr>
<td>BLG0B100 D/14 06 N/ 41/00/00 S0BEE</td>
</tr>
<tr>
<td>BLG6STAT A/ 2 06 B/U 00/04/00 S0BEE BC IMS0SSCC00 STAC/CLOSED</td>
</tr>
<tr>
<td>BLG6TPY A/ 2 06 B/U 41/00/00 S0C09</td>
</tr>
<tr>
<td>BLG6PTYP A/ 2 06 B/U 00/04/00 S0C00 BC IMGCT0000 TYPE/USER</td>
</tr>
<tr>
<td>BLG0B100 D/ 4 06 N/ 41/00/08 S0C3D</td>
</tr>
<tr>
<td>BLG6OCCD A/ 2 06 B/U 00/04/00 S0C3D BC IM00SD001 DATO/10/21/96</td>
</tr>
</tbody>
</table>

Figure 36. VIEW INTERNALS Output

DELENTRY

In order to delete an entry in a record, the s-word index associated with the
DELENTRY parameter in the BLGARCP data set must match the s-word index
specified with the data in the VIEW INTERNALS output for the record.

When choosing which s-word index to specify, remember the following rules:

- DELENTRY will only delete entries that contain data or a visible phrase.
- DELENTRY cannot delete data for which the associated entry in the VIEW
  INTERNALS output has an s-word index of x'0000'.
In Figure 37 on page 15-30, the lines marked by an asterisk may be deleted using the **DELENTRY** parameter.

<table>
<thead>
<tr>
<th>PANEL</th>
<th>PANEL REL COG- FLAGS</th>
<th>SWORD STRUCTURED PREFIX WORD OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>TYP/RSP LEV NIZE F M D</td>
<td>INDEX WORD VISIBLE PHRASE</td>
</tr>
<tr>
<td>*</td>
<td>BLG0EN20  D/ 5 06 B/ 19/00/88 S002C BA //S/TSI ENTRY</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SDDSROOT C/ 00 B/ 12/04/10 S0000 00003E44</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* BLG00000 S/ 1 06 B/ 18/04/00 S0032 BA //S/TXS RECS=PROBLEM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* BLG0B001 S/ 1 06 B/ 19/00/80 S00FC BC IM01AENT REPORTER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLG0B100 D/ 1 06 N/ 41/00/00 S0859</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* BLG6REQN A/ 2 06 B/U 00/04/00 S0859 BC IM01PS00 PERS/ANY/ONE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLG0B100 D/13 06 N/ 41/00/00 S0C09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLG6PTYP A/ 2 06 B/U 00/04/00 S0000 BC IMGCT0000 TYPE/USER</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLG0B100 D/14 06 N/ 41/00/00 S0BEE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* BLG6STAT A/ 2 06 B/U 00/04/00 S0BEE BC IMS05SC00 STAC/CLOSED</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLG0B100 D/25 06 N/ 41/00/00 S0E0F</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* BLG6SAB A/ 2 06 B/ 00/04/00 S0E0F BC IM0TXCA00 TEST</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BLG0B100 D/ 4 06 N/ 41/00/08 S0C3D</td>
<td></td>
</tr>
<tr>
<td></td>
<td>* BLG6CCD A/ 2 06 B/U 00/04/00 S0C3D BC IM00SD001 DATO/10/21/96</td>
<td></td>
</tr>
</tbody>
</table>

Figure 37. VIEW INTERNALS Output before DELENTRY

Thus, if you wanted to delete date occurred (DATO/10/21/96), you could specify:

**DELENTRY=S0C3D**

and then the **VIEW INTERNALS** data of the archived record would look like:

<table>
<thead>
<tr>
<th>PANEL</th>
<th>PANEL REL COG- FLAGS</th>
<th>SWORD STRUCTURED PREFIX WORD OR</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>TYP/RSP LEV NIZE F M D</td>
<td>INDEX WORD VISIBLE PHRASE</td>
</tr>
<tr>
<td>BLG0EN20  D/ 5 06 B/ 19/00/88 S002C BA //S/TSI ENTRY</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDDSROOT C/ 00 B/ 12/04/10 S0000 00003E44</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* BLG00000 S/ 1 06 B/ 18/04/00 S0032 BA //S/TXS RECS=PROBLEM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* BLG0B001 S/ 1 06 B/ 19/00/80 S00FC BC IM01AENT REPORTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100 D/ 1 06 N/ 41/00/00 S0859</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* BLG6REQN A/ 2 06 B/U 00/04/00 S0859 BC IM01PS00 PERS/ANY/ONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100 D/13 06 N/ 41/00/00 S0C09</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG6PTYP A/ 2 06 B/U 00/04/00 S0000 BC IMGCT0000 TYPE/USER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100 D/14 06 N/ 41/00/00 S0BEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* BLG6STAT A/ 2 06 B/U 00/04/00 S0BEE BC IMS05SC00 STAC/CLOSED</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100 D/25 06 N/ 41/00/00 S0E0F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>* BLG6SAB A/ 2 06 B/ 00/04/00 S0E0F BC IM0TXCA00 TEST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100 D/ 4 06 N/ 41/00/08 S0C3D</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 38. VIEW INTERNALS Output after DELENTRY=S0C3D

Notice that the line with s-word index x'0C3D' which contains no data is not deleted. For information on how to delete this line, refer to “COMPRESS” on page 15-33.
In the VIEW INTERNALS output in Figure 37, the data TYPE/USER cannot be deleted because the s-word index on the line with the data is X'0000'. Notice that the lines of data in the VIEW INTERNALS output are paired when the data is collected via an assisted-entry panel. That is, there is a line for the data-entry panel and for the assisted-entry panel for each piece of data. When the assisted-entry panel specifies an s-word, the VIEW INTERNALS line for the data will contain an s-word index other than X'0000'. However, when the assisted-entry panel specifies collect from caller, the s-word index associated with the data will be X'0000'. For additional information on how data is collected via an assisted-entry panel, refer to the Panel Modification Facility Guide and Reference.

Direct-add items (such as items added to a record with a Control panel ADD) only create 1 entry in the VIEW INTERNALS output.

**UNCATEGORIZE**

In order to uncategorize an entry in a record, the s-word index associated with the UNCOGNIZE parameter in the BLGARCP data set must match the s-word index specified with the data in the VIEW INTERNALS output for the record.

When UNCOGNIZE is specified, both the s-word and the p-word for the entry are uncategorized.

When choosing which s-word index to specify, remember the following rules:

- **UNCATEGORIZE** will only uncategorize entries that contain data or a visible phrase.
- **UNCATEGORIZE** cannot uncategorize entries for which the associated entry in the VIEW INTERNALS output has an s-word index of x'0000'.

Notice that the VIEW INTERNALS output marks which lines are categorized with the COGNIZE column. If the COGNIZE column has N/, the line is not categorized, and there is no reason to specify UNCOGNIZE to uncategorize it. If the line contains B/ or B/U, the line is categorized. If the line contains P/M, categorize only p-word is specified on the assisted-entry panel (additional information on these fields can be found in Terminal Simulator Guide and Reference). In Figure 39 on page 15-32, the lines marked by an asterisk may be uncategorized using the UNCOGNIZE parameter.
Thus, if you wanted to uncognize the status (STAC/CLOSED) because all of the records on your archive database have a status of closed, you could specify:

```
UNCOGNIZE=S0BEE
```

and then the **VIEW INTERNALS** data of the archived record would look like:

```
<table>
<thead>
<tr>
<th>PANEL</th>
<th>PANEL</th>
<th>REL COG-</th>
<th>FLAGS</th>
<th>SWORD</th>
<th>STRUCTURED</th>
<th>PREFIX WORD OR NAME</th>
<th>TYP/RSP</th>
<th>LEV</th>
<th>NIZE</th>
<th>F M D</th>
<th>INDEX</th>
<th>WORD VISIBLE PHRASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0EN20</td>
<td>D/ 5 06 B/ 19/00/88 S002C BA //S/TSI ENTRY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SDDSROOT</td>
<td>C/ 00 N/ 12/04/10 S0000 0000050D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG00000</td>
<td>S/ 06 B/ 18/04/00 S0032 BA //S/TXS RECS=PROBLEM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B001</td>
<td>S/ 06 B/ 19/00/80 SOCFC BC IMIAIENTO REPORTER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100</td>
<td>D/ 1 06 N/ 41/00/00 S0B59</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG6REQN</td>
<td>A/ 2 06 B/U 00/04/00 S0B59 BC IM0IOP0S0 PERS/ANY/ONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100</td>
<td>D/13 06 N/ 41/00/00 S0C09</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG6PTYP</td>
<td>A/ 2 06 B/U 00/04/00 S0000 BC IMGCT0000 TYPE/USER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100</td>
<td>D/14 06 N/ 41/00/00 S0BEE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG6STAT</td>
<td>A/ 2 06 B/U 00/04/00 S0BEE BC IM0S0S0C0 STAC/CLOSED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100</td>
<td>D/25 06 N/ 41/00/00 S0E0F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG6SAB</td>
<td>A/ 2 06 B/ 00/04/00 S0E0F BC IM0TXC0A TEST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0B100</td>
<td>D/ 4 06 N/ 41/00/08 S0C3D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG6OCCD</td>
<td>A/ 2 06 B/U 00/04/00 S0C3D BC IM00SD001 DATO/10/21/96</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 39. VIEW INTERNALS Output before UNCOGNIZE

Figure 40. VIEW INTERNALS Output after UNCOGNIZE=S0BEE
COMPRESS
In order to remove non-essential entries from a record, specify the COMPRESS=Y parameter in the BLGARCP data set. The following rules explain which entries will be removed with COMPRESS=Y:

- COMPRESS will remove all entries that do not contain data even if they contain an s-word index.
- COMPRESS will not remove entries that contain data or a visible phrase.

In Figure 41, the lines marked by an asterisk may be compressed using the COMPRESS parameter.

<table>
<thead>
<tr>
<th>BLG1TVID</th>
<th>VIEW INTERNAL DATA</th>
<th>LINE 1 OF 16</th>
</tr>
</thead>
<tbody>
<tr>
<td>PANEL</td>
<td>PANEL REL COG- FLAGS</td>
<td>STRUCTURED PREFIX WORD OR</td>
</tr>
<tr>
<td>NAME</td>
<td>TYP/RSP LEV NIZE F M D INDEX</td>
<td>WORD VISIBLE PHRASE</td>
</tr>
<tr>
<td>BLG6DENO</td>
<td>D/ 5 06 B/ 19/00/88 S002C BA //S/TSI ENTRY</td>
<td></td>
</tr>
<tr>
<td>SDDSROOT</td>
<td>C/ 00 N/ 12/04/10 S0000 000005DD</td>
<td></td>
</tr>
<tr>
<td>BLG00000</td>
<td>S/ 1 06 B/ 18/04/00 S0032 BA //S/TXS RECS=PROBLEM</td>
<td></td>
</tr>
<tr>
<td>BLG09801</td>
<td>S/ 1 06 B/ 19/00/80 S0CFC BC IMDIAENT0 REPORTER</td>
<td></td>
</tr>
<tr>
<td>* BLG0B100</td>
<td>D/ 1 06 N/ 41/00/00 S0B59</td>
<td></td>
</tr>
<tr>
<td>BLG6REQO</td>
<td>A/ 2 06 B/U 00/04/00 S0B59 BC IM0I0PS00 PER/ANY/ONE</td>
<td></td>
</tr>
<tr>
<td>* BLG0B100</td>
<td>D/13 06 N/ 41/00/00 S0C09</td>
<td></td>
</tr>
<tr>
<td>BLG6PTYP</td>
<td>A/ 2 06 B/U 00/04/00 S0000 BC IMGCT0000 TYPE/USER</td>
<td></td>
</tr>
<tr>
<td>* BLG0B100</td>
<td>D/14 06 N/ 41/00/00 S0BEE</td>
<td></td>
</tr>
<tr>
<td>BLG6STAT</td>
<td>A/ 2 06 B/U 00/04/00 S0BEE BC IMS05SC00 STAC/CLOSED</td>
<td></td>
</tr>
<tr>
<td>* BLG0B100</td>
<td>D/25 06 N/ 41/00/00 S0EOF</td>
<td></td>
</tr>
<tr>
<td>BLG6DSAB</td>
<td>A/ 2 06 B/ 00/04/00 S0EOF BC IM0TCA00 TEST</td>
<td></td>
</tr>
<tr>
<td>* BLG0B100</td>
<td>D/ 4 06 N/ 41/00/08 S0C3D</td>
<td></td>
</tr>
<tr>
<td>BLG6OCCD</td>
<td>A/ 2 06 B/U 00/04/00 S0C3D BC IM005D001 DATO/10/21/96</td>
<td></td>
</tr>
</tbody>
</table>

Figure 41. VIEW INTERNALS Output before COMPRESS

Thus, if you specify:

COMPRESS=Y

then the VIEW INTERNALS data of the archived record would look like:

<table>
<thead>
<tr>
<th>BLG1TVID</th>
<th>VIEW INTERNAL DATA</th>
<th>LINE 1 OF 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>PANEL</td>
<td>PANEL REL COG- FLAGS</td>
<td>STRUCTURED PREFIX WORD OR</td>
</tr>
<tr>
<td>NAME</td>
<td>TYP/RSP LEV NIZE F M D INDEX</td>
<td>WORD VISIBLE PHRASE</td>
</tr>
<tr>
<td>BLG6DENO</td>
<td>D/ 5 06 B/ 19/00/88 S002C BA //S/TSI ENTRY</td>
<td></td>
</tr>
<tr>
<td>SDDSROOT</td>
<td>C/ 00 N/ 12/04/10 S0000 000005DD</td>
<td></td>
</tr>
<tr>
<td>BLG00000</td>
<td>S/ 1 06 N/ 18/04/00 S0032 BA //S/TXS RECS=PROBLEM</td>
<td></td>
</tr>
<tr>
<td>BLG09801</td>
<td>S/ 1 06 N/ 19/00/80 S0CFC BC IMDIAENT0 REPORTER</td>
<td></td>
</tr>
<tr>
<td>BLG6REQO</td>
<td>A/ 2 06 B/U 00/04/00 S0B59 BC IM0I0PS00 PER/ANY/ONE</td>
<td></td>
</tr>
<tr>
<td>BLG6PTYP</td>
<td>A/ 2 06 B/U 00/04/00 S0000 BC IMGCT0000 TYPE/USER</td>
<td></td>
</tr>
<tr>
<td>BLG6STAT</td>
<td>A/ 2 06 B/U 00/04/00 S0BEE BC IMS05SC00 STAC/CLOSED</td>
<td></td>
</tr>
<tr>
<td>BLG6DSAB</td>
<td>A/ 2 06 N/ 00/04/00 S0EOF BC IM0TCA00 TEST</td>
<td></td>
</tr>
<tr>
<td>BLG6OCCD</td>
<td>A/ 2 06 B/U 00/04/00 S0C3D BC IM005D001 DATO/10/21/96</td>
<td></td>
</tr>
</tbody>
</table>

Figure 42. VIEW INTERNALS Output after COMPRESS=Y
Panel Processing Considerations

If you are planning to use the Archiver, you should also consider the following:

- You need to modify API create/update TSPs to enable special processing:
  - BLGAPI02 - Delete Line 16 which says BRANCH DISABLED.
  - BLGAPI05 - Delete Line 17 which says BRANCH DISABLED.
  - BLGAPI10 - Delete Line 9 which says BRANCH DISABLED.

  This should be done in the both the LIVESESS and ARCSESS sessions.

- The Archiver uses API “panel processing,” which means that the archived records are filed using file processing of the data-summary panels which were created. In some cases, the Archiver can use incorrect Create Summary panels when archiving “child”-type records (for example, change activity records). To avoid this situation, you can direct the API to use default file processing for archived records. As shipped, this default file processing performs only a record file and does not add information such as date/time last altered. To enable this for your archived records, do the following:
  - Identify the record types that you will be archiving.
  - For each record type that you want to file using the default file, modify the record's entry in control panel BLG1AACP to target summary panel BLG0DFSM. BLG1AACP contains a list of record type s-words and associated summary panels to use for filing the record. For child records, make sure that both the parent and child s-word rows target BLG0DFSM. Make sure also that the modified BLG1AACP is available to the ARCSESS session.

- If you do not choose to implement the default file processing panels, you must ensure that notification processing is disabled for the Archiver. If you use TSPs and TSO Send for notification, ensure that user exits are in place for the problem and/or change record file panels to prevent TSPs from running during API processing. If you use TSXs for e-mail notification, disable the calls to the TSXs in copies of the record file panels used by the API.
  - BLG1A111 - Problem Record File
  - BLG1A121 - Change Record File.

  Do this in the ARCSESS session.

- Only one record type can be archived/copied at a time. Use one of the following values for INQPIDT in the input parameters list:
  - BLGYACI - change activity records
  - BLGYCHI - change records
  - BLGYDCI - data center records
  - BLGYHCI - configuration hardware component records
  - BLGYHFI - configuration hardware component feature records
  - BLGYHNI - configuration financial hardware records
  - BLGYHSI - configuration hardware sub-component records
  - BLGYHXI - configuration hardware component connection
If you created your own record types you will need to create a unique inquiry PIDT to accommodate your records. Use the samples in SBLMSAMP as a model.

- Because the Archiver is a LLAPI job, it will be affected by the setting of the APISECURITY keyword in the BLX-SP parameters member. The APISECURITY keyword specifies whether to activate the BLX-SP security checking for APIs to validate that the MVS application user ID is allowed to access an Information/Management database. For more information on this keyword, refer to the Application Program Interface Guide.

**Logical Database Partition Considerations**

If you are using logical database partitioning, you must archive each separate logical partition of a database separately using the archiver. For each partition, the archive session must use a privilege class whose primary partition is the partition being archived. This ensures that the archived records are placed into the correct partition and that duplicate records will not be produced in the archive database if the same record ID is archived more than once.
Panel Lists

This appendix lists the entry, display, and inquiry panels that are shipped with TME 10 Information/Management Version 1 Release 1. “Information/Management Panels Arranged by Panel ID” lists the panels in alphanumeric order by panel identifier. “Information/Management Panels Arranged by Panel Title” on page A-20 lists the panels in alphabetic order by panel title.

Information/Management Panels Arranged by Panel ID

<table>
<thead>
<tr>
<th>Panel ID</th>
<th>Panel Description</th>
<th>Panel ID</th>
<th>Panel Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLGLALIS</td>
<td>Alias Name Entry</td>
<td>BLG0B100</td>
<td>Problem Reporter Entry</td>
</tr>
<tr>
<td>BLGLCMDD</td>
<td>Command Alias/Authorization Display</td>
<td>BLG0B200</td>
<td>Problem Status Entry</td>
</tr>
<tr>
<td></td>
<td>Command Alias/Authorization Entry</td>
<td>BLG0B300</td>
<td>Problem Close Entry</td>
</tr>
<tr>
<td>BLGLCMDU</td>
<td>Command Alias/Authorization Entry</td>
<td>BLG0B400</td>
<td>Problem Supplemental Entry</td>
</tr>
<tr>
<td>BLGLDATT</td>
<td>Data Attribute Records Entry</td>
<td>BLG0B402</td>
<td>Problem Symptom Data</td>
</tr>
<tr>
<td>BLGLJHPL</td>
<td>Sharing Logical Partition Identifier Entry</td>
<td>BLG0B500</td>
<td>Problem Synopsis Entry</td>
</tr>
<tr>
<td>BLGLJSPL</td>
<td>Authorized Logical Partition Identifier Entry</td>
<td>BLG0B502</td>
<td>Problem Resolution Data</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0B901</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0B902</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0B903</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU00</td>
<td>Change Request Summary</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU01</td>
<td>Change Request Summary</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU10</td>
<td>Change Delete Verification</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU11</td>
<td>Change Delete Verification</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU20</td>
<td>Activity Summary</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU30</td>
<td>Activity Delete Verification</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU31</td>
<td>Activity Delete Verification</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU03</td>
<td>Activity Entry</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU04</td>
<td>Change Entry</td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td>BLG0CU10</td>
<td>Change Text Entry</td>
</tr>
<tr>
<td>Panel ID</td>
<td>Description</td>
<td>Panel ID</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------</td>
<td>----------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>BLG0C030</td>
<td>Activity Text Entry</td>
<td>BLG0C163</td>
<td>Hardware Alter by Installing Inquiry</td>
</tr>
<tr>
<td>BLG0C040</td>
<td>Change Detail</td>
<td>BLG0C164</td>
<td>Hardware Alter by Modifying Inquiry</td>
</tr>
<tr>
<td>BLG0C050</td>
<td>Software Change</td>
<td>BLG0C165</td>
<td>Hardware Alter by Removing Inquiry</td>
</tr>
<tr>
<td>BLG0C051</td>
<td>Software Add</td>
<td>BLG0C166</td>
<td>Hardware Remove Inquiry</td>
</tr>
<tr>
<td>BLG0C052</td>
<td>Software Alter</td>
<td>BLG0C170</td>
<td>Document Add Inquiry</td>
</tr>
<tr>
<td>BLG0C053</td>
<td>Software Alter by Installing</td>
<td>BLG0C171</td>
<td>Document Alter Inquiry</td>
</tr>
<tr>
<td>BLG0C054</td>
<td>Software Alter by Modifying</td>
<td>BLG0C172</td>
<td>Document Remove Inquiry</td>
</tr>
<tr>
<td>BLG0C055</td>
<td>Software Alter by Removing</td>
<td>BLG0C173</td>
<td>Change Status Entry</td>
</tr>
<tr>
<td>BLG0C056</td>
<td>Software Remove</td>
<td>BLG0C174</td>
<td>Change Close Entry</td>
</tr>
<tr>
<td>BLG0C060</td>
<td>Hardware Change</td>
<td>BLG0C175</td>
<td>Change Detail</td>
</tr>
<tr>
<td>BLG0C061</td>
<td>Hardware Add</td>
<td>BLG0C176</td>
<td>Software Change</td>
</tr>
<tr>
<td>BLG0C062</td>
<td>Hardware Alter</td>
<td>BLG0C177</td>
<td>Add/Remove Software</td>
</tr>
<tr>
<td>BLG0C063</td>
<td>Hardware Alter by Installing</td>
<td>BLG0C178</td>
<td>Software Alteration</td>
</tr>
<tr>
<td>BLG0C064</td>
<td>Hardware Alter by Modifying</td>
<td>BLG0C179</td>
<td>Hardware Change</td>
</tr>
<tr>
<td>BLG0C065</td>
<td>Hardware Alter by Removing</td>
<td>BLG0C180</td>
<td>Add/Remove Hardware</td>
</tr>
<tr>
<td>BLG0C066</td>
<td>Hardware Remove</td>
<td>BLG0C181</td>
<td>Hardware Alteration</td>
</tr>
<tr>
<td>BLG0C070</td>
<td>Document Change</td>
<td>BLG0C182</td>
<td>Documentation Change</td>
</tr>
<tr>
<td>BLG0C071</td>
<td>Document Add</td>
<td>BLG0C183</td>
<td>Add/Remove Documentation</td>
</tr>
<tr>
<td>BLG0C072</td>
<td>Document Alter</td>
<td>BLG0C184</td>
<td>Entry Options</td>
</tr>
<tr>
<td>BLG0C073</td>
<td>Document Remove</td>
<td>BLG0C185</td>
<td>Software Level</td>
</tr>
<tr>
<td>BLG0C100</td>
<td>Change Requester Entry</td>
<td>BLG0C186</td>
<td>Software APAR</td>
</tr>
<tr>
<td>BLG0C151</td>
<td>Software Add Inquiry</td>
<td>BLG0C187</td>
<td>Software PTF</td>
</tr>
<tr>
<td>BLG0C152</td>
<td>Software Alter Inquiry</td>
<td>BLG0C188</td>
<td>Upgrade/Tape</td>
</tr>
<tr>
<td>BLG0C153</td>
<td>Software Alter by Installing Inquiry</td>
<td>BLG0C189</td>
<td>Upgrade Subsets</td>
</tr>
<tr>
<td>BLG0C154</td>
<td>Software Alter by Modifying Inquiry</td>
<td>BLG0C190</td>
<td>Installation Mods</td>
</tr>
<tr>
<td>BLG0C155</td>
<td>Software Alter by Removing Inquiry</td>
<td>BLG0C191</td>
<td>Engineering Change</td>
</tr>
<tr>
<td>BLG0C156</td>
<td>Software Remove Inquiry</td>
<td>BLG0C192</td>
<td>Device Card</td>
</tr>
<tr>
<td>BLG0C161</td>
<td>Hardware Add Inquiry</td>
<td>BLG0C193</td>
<td>Control Item</td>
</tr>
<tr>
<td>BLG0C162</td>
<td>Hardware Alter Inquiry</td>
<td>BLG0C194</td>
<td>Engineer Announcements</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0C195</td>
<td>Bill of Materials</td>
</tr>
<tr>
<td>Reference</td>
<td>Information/Management Panels Arranged by Panel ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C496</td>
<td>Optional Feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C497</td>
<td>Request Engineer Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C498</td>
<td>Request Price Quotation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C500</td>
<td>Change Approver Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C600</td>
<td>Activity Requester Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C620</td>
<td>Activity Status Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C630</td>
<td>Activity Close Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C640</td>
<td>Activity Detail</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C700</td>
<td>Change Reviewer Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DATT</td>
<td>Data Attribute Records Display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU01</td>
<td>Hardware Component Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU02</td>
<td>Hardware Feature Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU03</td>
<td>Software Component Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU04</td>
<td>Software Feature Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU05</td>
<td>Hardware Financial Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU06</td>
<td>Software Financial Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU07</td>
<td>Service Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU08</td>
<td>Data Center Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU09</td>
<td>System Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU10</td>
<td>Software Connection Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU11</td>
<td>Hardware Component Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU12</td>
<td>Hardware Feature Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU13</td>
<td>Software Component Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU14</td>
<td>Software Feature Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU15</td>
<td>Hardware Financial Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU16</td>
<td>Software Financial Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU17</td>
<td>Service Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU18</td>
<td>Center Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU19</td>
<td>System Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU20</td>
<td>Software Connection Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU21</td>
<td>Hardware Component Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU22</td>
<td>Hardware Feature Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU23</td>
<td>Software Component Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU24</td>
<td>Software Feature Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU25</td>
<td>Hardware Financial Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU26</td>
<td>Software Financial Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU27</td>
<td>Service Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU28</td>
<td>Center Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU29</td>
<td>System Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU30</td>
<td>Hardware Component Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU31</td>
<td>Software Component Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU32</td>
<td>Hardware Connection Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU33</td>
<td>Hardware Subcomponent Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU34</td>
<td>Hardware Connection Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU35</td>
<td>Hardware Connection Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU36</td>
<td>Model Hardware Component Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU37</td>
<td>Model Hardware Component Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU38</td>
<td>Model Hardware Feature Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU39</td>
<td>Model Hardware Subcomponent Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU40</td>
<td>Software Connection Summary</td>
<td>BLG0DU60</td>
<td>Hardware Subcomponent Rename Verification</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------</td>
<td>----------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>BLG0DU42</td>
<td>Hardware Subcomponent Delete Verification</td>
<td>BLG0DU61</td>
<td>Hardware Financial Rename Verification</td>
</tr>
<tr>
<td>BLG0DU43</td>
<td>Hardware Subcomponent Delete Verification</td>
<td>BLG0DU62</td>
<td>Software Component Rename Verification</td>
</tr>
<tr>
<td>BLG0DU44</td>
<td>Model Software Component Summary</td>
<td>BLG0DU63</td>
<td>Software Connection Rename Verification</td>
</tr>
<tr>
<td>BLG0DU45</td>
<td>Model Software Component Summary</td>
<td>BLG0DU64</td>
<td>Software Feature Rename Verification</td>
</tr>
<tr>
<td>BLG0DU46</td>
<td>Model Software Feature Summary</td>
<td>BLG0DU65</td>
<td>Software Financial Rename Verification</td>
</tr>
<tr>
<td>BLG0DU47</td>
<td>Model Hardware Component Delete Verification</td>
<td>BLG0DU66</td>
<td>Model Hardware Component Rename Verification</td>
</tr>
<tr>
<td>BLG0DU48</td>
<td>Model Hardware Component Delete Verification</td>
<td>BLG0DU67</td>
<td>Model Hardware Feature Rename Verification</td>
</tr>
<tr>
<td>BLG0DU49</td>
<td>Model Hardware Subcomponent Delete Verification</td>
<td>BLG0DU68</td>
<td>Model Hardware Subcomponent Rename Verification</td>
</tr>
<tr>
<td>BLG0DU50</td>
<td>Model Hardware Subcomponent Delete Verification</td>
<td>BLG0DU69</td>
<td>Model Software Component Rename Verification</td>
</tr>
<tr>
<td>BLG0DU51</td>
<td>Model Hardware Feature Delete Verification</td>
<td>BLG0DU70</td>
<td>Model Software Feature Rename Verification</td>
</tr>
<tr>
<td>BLG0DU52</td>
<td>Model Software Component Delete Verification</td>
<td>BLG0DU71</td>
<td>Center Rename Verification</td>
</tr>
<tr>
<td>BLG0DU53</td>
<td>Model Software Component Delete Verification</td>
<td>BLG0DU72</td>
<td>System Rename Verification</td>
</tr>
<tr>
<td>BLG0DU54</td>
<td>Model Hardware Feature Delete Verification</td>
<td>BLG0DU73</td>
<td>Service Rename Verification</td>
</tr>
<tr>
<td>BLG0DU55</td>
<td>Model Software Feature Delete Verification</td>
<td>BLG0DU74</td>
<td>Hardware Component Swap Verification</td>
</tr>
<tr>
<td>BLG0DU56</td>
<td>Model Software Feature Delete Verification</td>
<td>BLG0DU75</td>
<td>Hardware Subcomponent Swap Verification</td>
</tr>
<tr>
<td>BLG0DU57</td>
<td>Hardware Component Rename Verification</td>
<td>BLG0DU76</td>
<td>Software Component Swap Verification</td>
</tr>
<tr>
<td>BLG0DU58</td>
<td>Hardware Connection Rename Verification</td>
<td>BLG0D000</td>
<td>Configuration Entry</td>
</tr>
<tr>
<td>BLG0DU59</td>
<td>Hardware Feature Rename Verification</td>
<td>BLG0D010</td>
<td>Text Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0D020</td>
<td>Text Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0D025</td>
<td>Text Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0D030</td>
<td>Text Entry</td>
</tr>
<tr>
<td>Panel ID</td>
<td>Description</td>
<td>Panel ID</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>BLG0D040</td>
<td>Text Entry</td>
<td>BLG0D160</td>
<td>Hardware Subcomponent Support Entry</td>
</tr>
<tr>
<td>BLG0D050</td>
<td>Hardware Financial Text Entry</td>
<td>BLG0D164</td>
<td>Hardware Subcomponent Financial Entry</td>
</tr>
<tr>
<td>BLG0D055</td>
<td>Software Financial Text Entry</td>
<td>BLG0D170</td>
<td>Hardware Component Line Entry</td>
</tr>
<tr>
<td>BLG0D060</td>
<td>Text Entry</td>
<td>BLG0D172</td>
<td>Hardware Component Loop Entry</td>
</tr>
<tr>
<td>BLG0D090</td>
<td>Software Feature Text Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D095</td>
<td>Software Connection Text Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D100</td>
<td>Hardware Component Entry</td>
<td>BLG0D200</td>
<td>Software Component Entry</td>
</tr>
<tr>
<td>BLG0D101</td>
<td>Component Entry</td>
<td>BLG0D201</td>
<td>Software Component Entry</td>
</tr>
<tr>
<td>BLG0D102</td>
<td>Hardware Component Entry</td>
<td>BLG0D202</td>
<td>Software Component Entry</td>
</tr>
<tr>
<td>BLG0D103</td>
<td>Hardware Feature Entry</td>
<td>BLG0D203</td>
<td>Software Feature Entry</td>
</tr>
<tr>
<td>BLG0D104</td>
<td>Hardware Component Entry</td>
<td>BLG0D204</td>
<td>Specification for Software Model Copy</td>
</tr>
<tr>
<td>BLG0D105</td>
<td>Hardware Component Source Definition</td>
<td>BLG0D205</td>
<td>Software Component Source Definition Entry</td>
</tr>
<tr>
<td>BLG0D106</td>
<td>Hardware Connectivity Entry</td>
<td>BLG0D206</td>
<td>Software Connection Entry</td>
</tr>
<tr>
<td>BLG0D107</td>
<td>Specification for Hardware Model Copy</td>
<td>BLG0D220</td>
<td>Software Feature Entry</td>
</tr>
<tr>
<td>BLG0D109</td>
<td>Component Entry</td>
<td>BLG0D230</td>
<td>Software Component Connection Entry</td>
</tr>
<tr>
<td>BLG0D110</td>
<td>Component Entry</td>
<td>BLG0D250</td>
<td>Software Component Support Entry</td>
</tr>
<tr>
<td>BLG0D113</td>
<td>Hardware Component Diagram Map Data Entry</td>
<td>BLG0D251</td>
<td>Software Component Secondary Support Entry</td>
</tr>
<tr>
<td>BLG0D120</td>
<td>Hardware Feature Entry</td>
<td>BLG0D252</td>
<td>Software Component Financial Entry</td>
</tr>
<tr>
<td>BLG0D130</td>
<td>Hardware Connection Entry</td>
<td>BLG0D253</td>
<td>Software Component Financial Entry</td>
</tr>
<tr>
<td>BLG0D140</td>
<td>Hardware Subcomponent Entry</td>
<td>BLG0D254</td>
<td>Software Feature Financial Entry</td>
</tr>
<tr>
<td>BLG0D141</td>
<td>Subcomponent Entry</td>
<td>BLG0D257</td>
<td>Software Feature Support Entry</td>
</tr>
<tr>
<td>BLG0D150</td>
<td>Hardware Component Primary Support Entry</td>
<td>BLG0D300</td>
<td>Hardware Financial Entry</td>
</tr>
<tr>
<td>BLG0D151</td>
<td>Hardware Component Secondary Support Entry</td>
<td>BLG0D301</td>
<td>Hardware Financial Entry</td>
</tr>
<tr>
<td>BLG0D152</td>
<td>Hardware Component Financial Entry</td>
<td>BLG0D302</td>
<td>Hardware Financial Entry</td>
</tr>
<tr>
<td>BLG0D153</td>
<td>Hardware Component Financial Entry</td>
<td>BLG0D304</td>
<td>Hardware Financial Entry</td>
</tr>
<tr>
<td>BLG0D154</td>
<td>Hardware Feature Financial Entry</td>
<td>BLG0D305</td>
<td>Hardware Financial - Purchase Entry</td>
</tr>
<tr>
<td>BLG0D157</td>
<td>Hardware Feature Support Entry</td>
<td>BLG0D306</td>
<td>Hardware Financial - Rental/Lease Entry</td>
</tr>
<tr>
<td>Panel ID</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D307</td>
<td>Hardware Financial - VPA Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D308</td>
<td>Hardware Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D320</td>
<td>Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D350</td>
<td>Software Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D351</td>
<td>Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D352</td>
<td>Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D353</td>
<td>Software Financial - Basic License Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D354</td>
<td>Financial Entry from Record Copy</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D355</td>
<td>Software Financial - DSLO License Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D356</td>
<td>Software Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D357</td>
<td>Software Financial - VLA Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D400</td>
<td>Service Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D401</td>
<td>Service Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D500</td>
<td>Data Center Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D501</td>
<td>Data Center Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D520</td>
<td>Text Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D600</td>
<td>System Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D601</td>
<td>System Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D620</td>
<td>Text Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D700</td>
<td>Model Hardware Component Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D701</td>
<td>Model Create Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D702</td>
<td>Model Component Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D703</td>
<td>Model Hardware Component Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D704</td>
<td>Model Hardware Feature Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D705</td>
<td>Model Subcomponent Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D706</td>
<td>Model Component Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D707</td>
<td>Model Software Component Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D708</td>
<td>Model Component Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D709</td>
<td>Model Software Feature Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D720</td>
<td>Model Hardware Feature Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D740</td>
<td>Model Hardware Subcomponent Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D750</td>
<td>Model Hardware Component Primary Support Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D751</td>
<td>Model Hardware Component Secondary Support Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D752</td>
<td>Model Hardware Component Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D753</td>
<td>Model Hardware Component Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D754</td>
<td>Model Hardware Feature Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D755</td>
<td>Model Hardware Feature Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D756</td>
<td>Model Software Component Primary Support Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D757</td>
<td>Model Software Component Secondary Support Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D758</td>
<td>Model Software Component Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D759</td>
<td>Model Software Component Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D760</td>
<td>Model Hardware Subcomponent Primary Support Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D761</td>
<td>Model Hardware Subcomponent Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D762</td>
<td>Model Hardware Subcomponent Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D770</td>
<td>Model Component Line Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D772</td>
<td>Model Component Loop Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D780</td>
<td>Model Software Component Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D781</td>
<td>Model Software Component Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D783</td>
<td>Model Software Component Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D784</td>
<td>Model Software Feature Financial Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D787</td>
<td>Model Software Feature Support Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D790</td>
<td>Model Software Feature Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0EN20</td>
<td>Primary Options Menu</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Version 1.1
<table>
<thead>
<tr>
<th>Reference</th>
<th>Information/Management Panels Arranged by Panel ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0E090</td>
<td>Problem Inquiry Summary</td>
</tr>
<tr>
<td>BLG0E100</td>
<td>Problem Reporter</td>
</tr>
<tr>
<td>BLG0E101</td>
<td>Problem Reporter</td>
</tr>
<tr>
<td>BLG0E102</td>
<td>Problem Reporter</td>
</tr>
<tr>
<td>BLG0E190</td>
<td>Problem Reporter Data Inquiry</td>
</tr>
<tr>
<td>BLG0E200</td>
<td>Problem Status</td>
</tr>
<tr>
<td>BLG0E201</td>
<td>Problem Status</td>
</tr>
<tr>
<td>BLG0E202</td>
<td>Problem Status</td>
</tr>
<tr>
<td>BLG0E290</td>
<td>Problem Status Data Inquiry</td>
</tr>
<tr>
<td>BLG0E300</td>
<td>Problem Close</td>
</tr>
<tr>
<td>BLG0E301</td>
<td>Problem Close</td>
</tr>
<tr>
<td>BLG0E390</td>
<td>Problem Close Data Inquiry</td>
</tr>
<tr>
<td>BLG0E400</td>
<td>Problem Supplemental Data</td>
</tr>
<tr>
<td>BLG0E401</td>
<td>Problem Supplementary Data</td>
</tr>
<tr>
<td>BLG0E402</td>
<td>Problem Supplementary Data</td>
</tr>
<tr>
<td>BLG0E490</td>
<td>Problem Symptom Data Inquiry</td>
</tr>
<tr>
<td>BLG0E590</td>
<td>Problem Resolution Data Inquiry</td>
</tr>
<tr>
<td>BLG0E690</td>
<td>Problem Supplemental Data Inquiry</td>
</tr>
<tr>
<td>BLG0E700</td>
<td>Problem Record Control</td>
</tr>
<tr>
<td>BLG0E890</td>
<td>Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0F000</td>
<td>Change/Activity Inquiry</td>
</tr>
<tr>
<td>BLG0F090</td>
<td>Change Request Summary</td>
</tr>
<tr>
<td>BLG0F100</td>
<td>Change Inquiry</td>
</tr>
<tr>
<td>BLG0F110</td>
<td>Change Requester</td>
</tr>
<tr>
<td>BLG0F111</td>
<td>Change Requester</td>
</tr>
<tr>
<td>BLG0F120</td>
<td>Change Status</td>
</tr>
<tr>
<td>BLG0F121</td>
<td>Change Status</td>
</tr>
<tr>
<td>BLG0F122</td>
<td>Change Status</td>
</tr>
<tr>
<td>BLG0F130</td>
<td>Change Close</td>
</tr>
<tr>
<td>BLG0F131</td>
<td>Change Close</td>
</tr>
<tr>
<td>BLG0F150</td>
<td>Approver Classes</td>
</tr>
<tr>
<td>BLG0F170</td>
<td>Change/Activity Control</td>
</tr>
<tr>
<td>BLG0F171</td>
<td>Activity Control</td>
</tr>
<tr>
<td>BLG0F172</td>
<td>Change Control</td>
</tr>
<tr>
<td>BLG0F173</td>
<td>Change/Activity Control</td>
</tr>
<tr>
<td>BLG0F174</td>
<td>Activity Control</td>
</tr>
<tr>
<td>BLG0F175</td>
<td>Change Control</td>
</tr>
<tr>
<td>BLG0F190</td>
<td>Change Requester Inquiry</td>
</tr>
<tr>
<td>BLG0F200</td>
<td>Activity Inquiry</td>
</tr>
<tr>
<td>BLG0F210</td>
<td>Activity Requester</td>
</tr>
<tr>
<td>BLG0F211</td>
<td>Activity Requester</td>
</tr>
<tr>
<td>BLG0F220</td>
<td>Activity Status</td>
</tr>
<tr>
<td>BLG0F221</td>
<td>Activity Status</td>
</tr>
<tr>
<td>BLG0F222</td>
<td>Activity Status</td>
</tr>
<tr>
<td>BLG0F230</td>
<td>Activity Close</td>
</tr>
<tr>
<td>BLG0F231</td>
<td>Activity Close</td>
</tr>
<tr>
<td>BLG0F270</td>
<td>Change/Activity Request Summary</td>
</tr>
<tr>
<td>BLG0F271</td>
<td>Change/Activity Requester Data Inquiry</td>
</tr>
<tr>
<td>BLG0F272</td>
<td>Change/Activity Status Data Inquiry</td>
</tr>
<tr>
<td>BLG0F273</td>
<td>Change/Activity Close Data Inquiry</td>
</tr>
<tr>
<td>BLG0F278</td>
<td>Change/Activity Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0F280</td>
<td>Activity Summary</td>
</tr>
<tr>
<td>BLG0F281</td>
<td>Activity Requester Data Inquiry</td>
</tr>
<tr>
<td>BLG0F282</td>
<td>Activity Status Data Inquiry</td>
</tr>
<tr>
<td>BLG0F283</td>
<td>Activity Close Data Inquiry</td>
</tr>
<tr>
<td>BLG0F288</td>
<td>Activity Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0F290</td>
<td>Change Status Data Inquiry</td>
</tr>
<tr>
<td>BLG0F300</td>
<td>Change/Activity Inquiry</td>
</tr>
<tr>
<td>BLG0F310</td>
<td>Change/Activity Requester</td>
</tr>
<tr>
<td>Panel ID</td>
<td>Panel Name</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>BLG0F311</td>
<td>Change/Activity Requester</td>
</tr>
<tr>
<td>BLG0F320</td>
<td>Change/Activity Status</td>
</tr>
<tr>
<td>BLG0F321</td>
<td>Change/Activity Status</td>
</tr>
<tr>
<td>BLG0F322</td>
<td>Change/Activity Status</td>
</tr>
<tr>
<td>BLG0F330</td>
<td>Change/Activity Close</td>
</tr>
<tr>
<td>BLG0F331</td>
<td>Change/Activity Close</td>
</tr>
<tr>
<td>BLG0F390</td>
<td>Change Close Data Inquiry</td>
</tr>
<tr>
<td>BLG0F400</td>
<td>Change Detail</td>
</tr>
<tr>
<td>BLG0F410</td>
<td>Software Change</td>
</tr>
<tr>
<td>BLG0F411</td>
<td>Add/Remove Software</td>
</tr>
<tr>
<td>BLG0F412</td>
<td>Software Modification</td>
</tr>
<tr>
<td>BLG0F413</td>
<td>Software Changes+</td>
</tr>
<tr>
<td>BLG0F420</td>
<td>Hardware Change</td>
</tr>
<tr>
<td>BLG0F421</td>
<td>Add/Remove Hardware</td>
</tr>
<tr>
<td>BLG0F422</td>
<td>Device Modification</td>
</tr>
<tr>
<td>BLG0F423</td>
<td>Generic Hardware</td>
</tr>
<tr>
<td>BLG0F430</td>
<td>Documentation Change</td>
</tr>
<tr>
<td>BLG0F431</td>
<td>Documentation Change</td>
</tr>
<tr>
<td>BLG0F440</td>
<td>All Changes</td>
</tr>
<tr>
<td>BLG0F445</td>
<td>All Changes</td>
</tr>
<tr>
<td>BLG0F450</td>
<td>Change Detail</td>
</tr>
<tr>
<td>BLG0F481</td>
<td>Software Level</td>
</tr>
<tr>
<td>BLG0F482</td>
<td>Software APAR</td>
</tr>
<tr>
<td>BLG0F483</td>
<td>Software PTF</td>
</tr>
<tr>
<td>BLG0F484</td>
<td>Upgrade/Tape</td>
</tr>
<tr>
<td>BLG0F485</td>
<td>Upgrade Subsets</td>
</tr>
<tr>
<td>BLG0F486</td>
<td>Installation Mods</td>
</tr>
<tr>
<td>BLG0F491</td>
<td>Engineering Change</td>
</tr>
<tr>
<td>BLG0F492</td>
<td>Device Card</td>
</tr>
<tr>
<td>BLG0F493</td>
<td>Control Item</td>
</tr>
<tr>
<td>BLG0F494</td>
<td>Engineer Announcements</td>
</tr>
<tr>
<td>BLG0F495</td>
<td>Bill of Materials</td>
</tr>
<tr>
<td>Reference</td>
<td>Information/Management Panels Arranged by Panel ID</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>BLG0G113</td>
<td>Hardware Components (Secondary)</td>
</tr>
<tr>
<td>BLG0G115</td>
<td>Hardware Components</td>
</tr>
<tr>
<td>BLG0G120</td>
<td>Hardware Feature</td>
</tr>
<tr>
<td>BLG0G121</td>
<td>Hardware Feature</td>
</tr>
<tr>
<td>BLG0G125</td>
<td>Feature Inquiry</td>
</tr>
<tr>
<td>BLG0G126</td>
<td>Connection Inquiry</td>
</tr>
<tr>
<td>BLG0G127</td>
<td>Hardware Connection</td>
</tr>
<tr>
<td>BLG0G128</td>
<td>Hardware Connection</td>
</tr>
<tr>
<td>BLG0G130</td>
<td>Subcomponent Inquiry</td>
</tr>
<tr>
<td>BLG0G131</td>
<td>Hardware Subcomponent</td>
</tr>
<tr>
<td>BLG0G135</td>
<td>Hardware+ Inquiry</td>
</tr>
<tr>
<td>BLG0G136</td>
<td>Hardware Records+</td>
</tr>
<tr>
<td>BLG0G137</td>
<td>Hardware Records+ Inquiry</td>
</tr>
<tr>
<td>BLG0G138</td>
<td>Hardware Records+ Summary</td>
</tr>
<tr>
<td>BLG0G140</td>
<td>Software Component</td>
</tr>
<tr>
<td>BLG0G141</td>
<td>Software Component</td>
</tr>
<tr>
<td>BLG0G142</td>
<td>Software Component</td>
</tr>
<tr>
<td>BLG0G145</td>
<td>Software Components</td>
</tr>
<tr>
<td>BLG0G150</td>
<td>Software Feature</td>
</tr>
<tr>
<td>BLG0G151</td>
<td>Software Feature</td>
</tr>
<tr>
<td>BLG0G155</td>
<td>Feature Inquiry</td>
</tr>
<tr>
<td>BLG0G160</td>
<td>Connection Inquiry</td>
</tr>
<tr>
<td>BLG0G161</td>
<td>Software Connection</td>
</tr>
<tr>
<td>BLG0G165</td>
<td>Software+ Inquiry</td>
</tr>
<tr>
<td>BLG0G166</td>
<td>Software Records+</td>
</tr>
<tr>
<td>BLG0G167</td>
<td>Software Records+ Inquiry</td>
</tr>
<tr>
<td>BLG0G168</td>
<td>Software Records+ Summary</td>
</tr>
<tr>
<td>BLG0G170</td>
<td>Hardware Subcomponent Summary</td>
</tr>
<tr>
<td>BLG0G171</td>
<td>Hardware Subcomponent Description/Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0G172</td>
<td>Hardware Subcomponent Support/EC Levels/Financial Inquiry</td>
</tr>
<tr>
<td>BLG0G175</td>
<td>Feature+ Inquiry</td>
</tr>
<tr>
<td>BLG0G176</td>
<td>Feature Records+</td>
</tr>
<tr>
<td>BLG0G180</td>
<td>Connection+ Inquiry</td>
</tr>
<tr>
<td>BLG0G181</td>
<td>Connection Records+</td>
</tr>
<tr>
<td>BLG0G182</td>
<td>Hardware Feature Support/EC Levels/Financial Data Inquiry</td>
</tr>
<tr>
<td>BLG0G183</td>
<td>Hardware Feature Summary</td>
</tr>
<tr>
<td>BLG0G184</td>
<td>Hardware Feature Description/Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0G185</td>
<td>Component+ Inquiry</td>
</tr>
<tr>
<td>BLG0G186</td>
<td>Component Records+</td>
</tr>
<tr>
<td>BLG0G190</td>
<td>Hardware Component Inquiry Summary</td>
</tr>
<tr>
<td>BLG0G191</td>
<td>Hardware Primary Description/Support Data Inquiry</td>
</tr>
<tr>
<td>BLG0G192</td>
<td>Hardware Secondary Description/Support Data Inquiry</td>
</tr>
<tr>
<td>BLG0G193</td>
<td>Hardware Financial/Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0G195</td>
<td>Records+ Inquiry</td>
</tr>
<tr>
<td>BLG0G196</td>
<td>Component Records+</td>
</tr>
<tr>
<td>BLG0G290</td>
<td>System Inquiry</td>
</tr>
<tr>
<td>BLG0G293</td>
<td>System Summary</td>
</tr>
<tr>
<td>BLG0G300</td>
<td>Financial Inquiry</td>
</tr>
<tr>
<td>BLG0G310</td>
<td>Hardware Financial</td>
</tr>
<tr>
<td>BLG0G311</td>
<td>Hardware Financial</td>
</tr>
<tr>
<td>BLG0G312</td>
<td>Hardware Financial (Secondary)</td>
</tr>
<tr>
<td>BLG0G313</td>
<td>Hardware Financial (Secondary)</td>
</tr>
<tr>
<td>BLG0G315</td>
<td>Hardware Financial Inquiry</td>
</tr>
<tr>
<td>BLG0G316</td>
<td>Hardware Financial VPA</td>
</tr>
<tr>
<td>BLG0G335</td>
<td>Financial+ Inquiry</td>
</tr>
<tr>
<td>Panel ID</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>BLG0G336</td>
<td>Financial Records+</td>
</tr>
<tr>
<td>BLG0G337</td>
<td>All Financial+ Inquiry</td>
</tr>
<tr>
<td>BLG0G338</td>
<td>All Financial+ Summary</td>
</tr>
<tr>
<td>BLG0G350</td>
<td>Software Financial</td>
</tr>
<tr>
<td>BLG0G351</td>
<td>Software Financial</td>
</tr>
<tr>
<td>BLG0G355</td>
<td>Software Financial Inquiry</td>
</tr>
<tr>
<td>BLG0G357</td>
<td>Software Financial Description</td>
</tr>
<tr>
<td>BLG0G358</td>
<td>Software Financial Summary</td>
</tr>
<tr>
<td>BLG0G359</td>
<td>Software Financial VLA/Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0G360</td>
<td>Software Financial (Secondary)</td>
</tr>
<tr>
<td>BLG0G361</td>
<td>Software Financial (Secondary)</td>
</tr>
<tr>
<td>BLG0G362</td>
<td>Software Financial VLA</td>
</tr>
<tr>
<td>BLG0G363</td>
<td>Software Financial (Secondary)</td>
</tr>
<tr>
<td>BLG0G390</td>
<td>Hardware Financial Summary</td>
</tr>
<tr>
<td>BLG0G391</td>
<td>Hardware Financial Description</td>
</tr>
<tr>
<td>BLG0G392</td>
<td>Hardware Financial Secondary Description Data Inquiry</td>
</tr>
<tr>
<td>BLG0G400</td>
<td>Service Inquiry</td>
</tr>
<tr>
<td>BLG0G411</td>
<td>Service Inquiry</td>
</tr>
<tr>
<td>BLG0G490</td>
<td>Service Inquiry</td>
</tr>
<tr>
<td>BLG0G493</td>
<td>Service Summary</td>
</tr>
<tr>
<td>BLG0G500</td>
<td>Data Center Inquiry</td>
</tr>
<tr>
<td>BLG0G511</td>
<td>Center Inquiry</td>
</tr>
<tr>
<td>BLG0G590</td>
<td>Data Center Inquiry</td>
</tr>
<tr>
<td>BLG0G593</td>
<td>Data Center Summary</td>
</tr>
<tr>
<td>BLG0G700</td>
<td>Model Component Records</td>
</tr>
<tr>
<td>BLG0G710</td>
<td>Model Hardware Components</td>
</tr>
<tr>
<td>BLG0G711</td>
<td>Model Hardware Component</td>
</tr>
<tr>
<td>BLG0G712</td>
<td>Model Hardware Component</td>
</tr>
<tr>
<td>BLG0G713</td>
<td>Model Hardware Components</td>
</tr>
<tr>
<td>Reference</td>
<td>Information/Management Panels Arranged by Panel ID</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>BLG0G772</td>
<td>Model Software Component Other Data Inquiry</td>
</tr>
<tr>
<td>BLG0G775</td>
<td>Model Software Feature Inquiry</td>
</tr>
<tr>
<td>BLG0G776</td>
<td>Model Software Feature</td>
</tr>
<tr>
<td>BLG0G777</td>
<td>Model Software Feature</td>
</tr>
<tr>
<td>BLG0G780</td>
<td>Model Software Feature Summary</td>
</tr>
<tr>
<td>BLG0G781</td>
<td>Model Software Feature Description Data Inquiry</td>
</tr>
<tr>
<td>BLG0G782</td>
<td>Model Software Feature Support/Financial Data Inquiry</td>
</tr>
<tr>
<td>BLG0G785</td>
<td>Model Software+ Inquiry</td>
</tr>
<tr>
<td>BLG0G786</td>
<td>Model Software Records+</td>
</tr>
<tr>
<td>BLG0G787</td>
<td>Model Software Records+ Inquiry</td>
</tr>
<tr>
<td>BLG0G788</td>
<td>Model Software Records+ Summary</td>
</tr>
<tr>
<td>BLG0G790</td>
<td>Model Hardware Component Summary</td>
</tr>
<tr>
<td>BLG0G791</td>
<td>Model Hardware Description Data Inquiry</td>
</tr>
<tr>
<td>BLG0G792</td>
<td>Model Hardware Support/Financial/Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0G795</td>
<td>Model Records+ Inquiry</td>
</tr>
<tr>
<td>BLG0G796</td>
<td>Model Component Records+</td>
</tr>
<tr>
<td>BLG0G800</td>
<td>Record Control</td>
</tr>
<tr>
<td>BLG0G801</td>
<td>Record Control</td>
</tr>
<tr>
<td>BLG0G850</td>
<td>System Inquiry</td>
</tr>
<tr>
<td>BLG0G851</td>
<td>System Records</td>
</tr>
<tr>
<td>BLG0G890</td>
<td>Control Inquiry</td>
</tr>
<tr>
<td>BLG0G893</td>
<td>Control Summary</td>
</tr>
<tr>
<td>BLG0I0E0</td>
<td>APAR Display</td>
</tr>
<tr>
<td>BLG0I010</td>
<td>Rules Inquiry</td>
</tr>
<tr>
<td>BLG0I020</td>
<td>Rules Inquiry</td>
</tr>
<tr>
<td>BLG0I030</td>
<td>Rules Inquiry</td>
</tr>
<tr>
<td>BLG0I040</td>
<td>Rules Record Control</td>
</tr>
<tr>
<td>BLG0I100</td>
<td>Escalation Rules Inquiry</td>
</tr>
<tr>
<td>BLG0I200</td>
<td>Escalation Control Inquiry</td>
</tr>
<tr>
<td>BLG0J000</td>
<td>Class Summary</td>
</tr>
<tr>
<td>BLG0J010</td>
<td>Class Delete Verification</td>
</tr>
<tr>
<td>BLG0J011</td>
<td>Class Delete Verification</td>
</tr>
<tr>
<td>BLG0J001</td>
<td>Privilege Class Entry</td>
</tr>
<tr>
<td>BLG0J010</td>
<td>Privilege Class Text Entry</td>
</tr>
<tr>
<td>BLG0J100</td>
<td>Class Description Entry</td>
</tr>
<tr>
<td>BLG0J200</td>
<td>Problem Authority</td>
</tr>
<tr>
<td>BLG0J210</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BLG0J211</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BLG0J212</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BLG0J220</td>
<td>Change Authority</td>
</tr>
<tr>
<td>BLG0J230</td>
<td>Configuration Authority</td>
</tr>
<tr>
<td>BLG0J240</td>
<td>Financial Authority</td>
</tr>
<tr>
<td>BLG0J250</td>
<td>Escalation Rules Authority</td>
</tr>
<tr>
<td>BLG0J260</td>
<td>Class Authority</td>
</tr>
<tr>
<td>BLG0J270</td>
<td>PMF Authority</td>
</tr>
<tr>
<td>BLG0J290</td>
<td>SRC Authority</td>
</tr>
<tr>
<td>BLG0J300</td>
<td>Class User Entry</td>
</tr>
<tr>
<td>BLG0K000</td>
<td>Class Inquiry</td>
</tr>
<tr>
<td>BLG0K091</td>
<td>Class Summary</td>
</tr>
<tr>
<td>BLG0K100</td>
<td>Class Description</td>
</tr>
<tr>
<td>BLG0K190</td>
<td>Class Description Inquiry</td>
</tr>
<tr>
<td>BLG0K2A0</td>
<td>Record Authority</td>
</tr>
<tr>
<td>BLG0K200</td>
<td>Problem Authority</td>
</tr>
<tr>
<td>BLG0K210</td>
<td>Authority Inquiry</td>
</tr>
<tr>
<td>BLG0K211</td>
<td>Authority Inquiry</td>
</tr>
<tr>
<td>BLG0K212</td>
<td>Authority Inquiry</td>
</tr>
<tr>
<td>BLG0K220</td>
<td>Change Authority</td>
</tr>
<tr>
<td>Panel ID</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>BLG0K230</td>
<td>Configuration Authority</td>
</tr>
<tr>
<td>BLG0K240</td>
<td>Financial Authority</td>
</tr>
<tr>
<td>BLG0K250</td>
<td>Escalation Rules Authority</td>
</tr>
<tr>
<td>BLG0K260</td>
<td>Class Authority</td>
</tr>
<tr>
<td>BLG0K270</td>
<td>PMF Authority</td>
</tr>
<tr>
<td>BLG0K290</td>
<td>SRC Authority</td>
</tr>
<tr>
<td>BLG0K291</td>
<td>Management Class Authority</td>
</tr>
<tr>
<td>BLG0K292</td>
<td>System Class Authority</td>
</tr>
<tr>
<td>BLG0K293</td>
<td>Record Authority</td>
</tr>
<tr>
<td>BLG0K800</td>
<td>Record Control</td>
</tr>
<tr>
<td>BLG0L010</td>
<td>Display Problem Text</td>
</tr>
<tr>
<td>BLG0L100</td>
<td>Problem Reporter Display</td>
</tr>
<tr>
<td>BLG0L200</td>
<td>Problem Status Display</td>
</tr>
<tr>
<td>BLG0L300</td>
<td>Problem Close Display</td>
</tr>
<tr>
<td>BLG0L400</td>
<td>Problem Supplemental Display</td>
</tr>
<tr>
<td>BLG0L500</td>
<td>Problem Synopsis Display</td>
</tr>
<tr>
<td>BLG0L600</td>
<td>Interested Class Display</td>
</tr>
<tr>
<td>BLG0L901</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG0L902</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG0L903</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG0M010</td>
<td>Display Change Text</td>
</tr>
<tr>
<td>BLG0M019</td>
<td>Display Activity Text</td>
</tr>
<tr>
<td>BLG0M100</td>
<td>Change Requester Display</td>
</tr>
<tr>
<td>BLG0M200</td>
<td>Change Status Display</td>
</tr>
<tr>
<td>BLG0M300</td>
<td>Change Close Display</td>
</tr>
<tr>
<td>BLG0M500</td>
<td>Change Approver Display</td>
</tr>
<tr>
<td>BLG0M600</td>
<td>Activity Requester Display</td>
</tr>
<tr>
<td>BLG0M620</td>
<td>Activity Status Display</td>
</tr>
<tr>
<td>BLG0M630</td>
<td>Activity Close Display</td>
</tr>
<tr>
<td>BLG0M700</td>
<td>Change Reviewer Display</td>
</tr>
<tr>
<td>BLG0N010</td>
<td>Display Hardware Text</td>
</tr>
<tr>
<td>BLG0N020</td>
<td>Display Feature Text</td>
</tr>
<tr>
<td>BLG0N142</td>
<td>Hardware Subcomponent Display</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>BLG0N150</td>
<td>Hardware Component Primary Support Display</td>
</tr>
<tr>
<td>BLG0N151</td>
<td>Hardware Component Secondary Support Display</td>
</tr>
<tr>
<td>BLG0N153</td>
<td>Hardware Component Financial Display</td>
</tr>
<tr>
<td>BLG0N154</td>
<td>Hardware Feature Financial Display</td>
</tr>
<tr>
<td>BLG0N157</td>
<td>Hardware Feature Support Display</td>
</tr>
<tr>
<td>BLG0N160</td>
<td>Hardware Subcomponent Support Display</td>
</tr>
<tr>
<td>BLG0N164</td>
<td>Hardware Subcomponent Financial Display</td>
</tr>
<tr>
<td>BLG0N170</td>
<td>Hardware Component Line Display</td>
</tr>
<tr>
<td>BLG0N172</td>
<td>Hardware Component Loop Display</td>
</tr>
<tr>
<td>BLG0N200</td>
<td>Software Component Summary Display</td>
</tr>
<tr>
<td>BLG0N201</td>
<td>Software Component Display</td>
</tr>
<tr>
<td>BLG0N205</td>
<td>Software Component Source Definition Display</td>
</tr>
<tr>
<td>BLG0N210</td>
<td>Software Component Record Utilities</td>
</tr>
<tr>
<td>BLG0N220</td>
<td>Software Feature Summary Display</td>
</tr>
<tr>
<td>BLG0N221</td>
<td>Software Feature Record Utilities</td>
</tr>
<tr>
<td>BLG0N222</td>
<td>Software Feature Display</td>
</tr>
<tr>
<td>BLG0N230</td>
<td>Software Connection Summary Display</td>
</tr>
<tr>
<td>BLG0N231</td>
<td>Software Connection Record Utilities</td>
</tr>
<tr>
<td>BLG0N232</td>
<td>Software Connection Display</td>
</tr>
<tr>
<td>BLG0N250</td>
<td>Software Component Support Display</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Panel ID</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>BLG0N700</td>
<td>Model Hardware Component Summary Display</td>
</tr>
<tr>
<td>BLG0N701</td>
<td>Model Hardware Component Display</td>
</tr>
<tr>
<td>BLG0N710</td>
<td>Model Hardware Component Record Utilities</td>
</tr>
<tr>
<td>BLG0N720</td>
<td>Model Hardware Feature Summary Display</td>
</tr>
<tr>
<td>BLG0N721</td>
<td>Model Hardware Feature Record Utilities</td>
</tr>
<tr>
<td>BLG0N722</td>
<td>Model Hardware Feature Display</td>
</tr>
<tr>
<td>BLG0N740</td>
<td>Model Hardware Subcomponent Display</td>
</tr>
<tr>
<td>BLG0N741</td>
<td>Model Hardware Subcomponent Record Utilities</td>
</tr>
<tr>
<td>BLG0N742</td>
<td>Model Hardware Subcomponent Display</td>
</tr>
<tr>
<td>BLG0N750</td>
<td>Model Hardware Component Primary Support Display</td>
</tr>
<tr>
<td>BLG0N751</td>
<td>Model Hardware Component Secondary Support Display</td>
</tr>
<tr>
<td>BLG0N753</td>
<td>Model Hardware Component Financial Display</td>
</tr>
<tr>
<td>BLG0N754</td>
<td>Model Hardware Feature Financial Display</td>
</tr>
<tr>
<td>BLG0N755</td>
<td>Model Hardware Feature Financial Display</td>
</tr>
<tr>
<td>BLG0N756</td>
<td>Model Hardware Feature Support Display</td>
</tr>
<tr>
<td>BLG0N757</td>
<td>Model Hardware Feature Support Display</td>
</tr>
<tr>
<td>BLG0N758</td>
<td>Model Software Component Primary Support Display</td>
</tr>
<tr>
<td>BLG0N759</td>
<td>Model Software Component Secondary Support Display</td>
</tr>
<tr>
<td>BLG0N760</td>
<td>Model Hardware Subcomponent Support Display</td>
</tr>
<tr>
<td>BLG0N764</td>
<td>Model Hardware Subcomponent Financial Display</td>
</tr>
<tr>
<td>BLG0N770</td>
<td>Model Hardware Component Line Display</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Panel ID</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>BLG0P511</td>
<td>Standard Report SYSOUT Destination Entry</td>
</tr>
<tr>
<td>BLG0P520</td>
<td>Standard Report Data Set Destination Entry</td>
</tr>
<tr>
<td>BLG0P521</td>
<td>Standard Report Data Set Destination Entry</td>
</tr>
<tr>
<td>BLG0P530</td>
<td>Standard Report DDNAME Destination Entry</td>
</tr>
<tr>
<td>BLG0P531</td>
<td>Standard Report DDNAME Destination Entry</td>
</tr>
<tr>
<td>BLG0P700</td>
<td>Data Definition Defaults</td>
</tr>
<tr>
<td>BLG0P800</td>
<td>Draw Report Destination</td>
</tr>
<tr>
<td>BLG0P801</td>
<td>Draw Report Output Destination</td>
</tr>
<tr>
<td>BLG0P802</td>
<td>Draw Report Output Destination</td>
</tr>
<tr>
<td>BLG0P810</td>
<td>Draw Report SYSOUT Destination Entry</td>
</tr>
<tr>
<td>BLG0P811</td>
<td>Draw Report SYSOUT Destination Entry</td>
</tr>
<tr>
<td>BLG0P820</td>
<td>Draw Report Data Set Destination Entry</td>
</tr>
<tr>
<td>BLG0P821</td>
<td>Draw Report Data Set Destination Entry</td>
</tr>
<tr>
<td>BLG0P830</td>
<td>Draw Report DDNAME Destination Entry</td>
</tr>
<tr>
<td>BLG0P831</td>
<td>Draw Report DDNAME Destination Entry</td>
</tr>
<tr>
<td>BLG0QUP0</td>
<td>Record Utilities</td>
</tr>
<tr>
<td>BLG0Q010</td>
<td>Display Privilege Class Text</td>
</tr>
<tr>
<td>BLG0Q100</td>
<td>Class Summary Display</td>
</tr>
<tr>
<td>BLG0Q200</td>
<td>Problem Authority</td>
</tr>
<tr>
<td>BLG0Q210</td>
<td>Authority Display</td>
</tr>
<tr>
<td>BLG0Q211</td>
<td>Authority Display</td>
</tr>
<tr>
<td>BLG0Q212</td>
<td>Authority Display</td>
</tr>
<tr>
<td>BLG0Q213</td>
<td>Authority Summary</td>
</tr>
<tr>
<td>BLG0Q214</td>
<td>Authority Summary</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Panel ID</td>
<td>Panel Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>BLG0T100</td>
<td>SRC Summary Display</td>
</tr>
<tr>
<td>BLG0T101</td>
<td>Record Utilities</td>
</tr>
<tr>
<td>BLG0T110</td>
<td>SRC Description Display</td>
</tr>
<tr>
<td>BLG0T120</td>
<td>Final Responses Display</td>
</tr>
<tr>
<td>BLG0T130</td>
<td>Display SRC Responses</td>
</tr>
<tr>
<td>BLG0T200</td>
<td>SRC Execution Classes</td>
</tr>
<tr>
<td>BLG0U000</td>
<td>SRC Summary</td>
</tr>
<tr>
<td>BLG0U001</td>
<td>SRC Summary</td>
</tr>
<tr>
<td>BLG0U100</td>
<td>SRC Description Entry</td>
</tr>
<tr>
<td>BLG0U101</td>
<td>SRC Entry</td>
</tr>
<tr>
<td>BLG0U120</td>
<td>Final SRC Response Entry</td>
</tr>
<tr>
<td>BLG0U200</td>
<td>SRC Execution Classes</td>
</tr>
<tr>
<td>BLG0VD50</td>
<td>Validation Record Delete Verification</td>
</tr>
<tr>
<td>BLG0VD51</td>
<td>Validation Record Delete Verification</td>
</tr>
<tr>
<td>BLG0VD60</td>
<td>Data View Record Delete Verification</td>
</tr>
<tr>
<td>BLG0VD61</td>
<td>Data View Record Delete Verification</td>
</tr>
<tr>
<td>BLG0VD70</td>
<td>Data Attribute Record Delete Verification</td>
</tr>
<tr>
<td>BLG0VD71</td>
<td>Data Attribute Record Delete Verification</td>
</tr>
<tr>
<td>BLG0VE60</td>
<td>Data View Inquiry Summary</td>
</tr>
<tr>
<td>BLG0VE61</td>
<td>Data View Record Inquiry</td>
</tr>
<tr>
<td>BLG0VE62</td>
<td>Authorization Code Inquiry</td>
</tr>
<tr>
<td>BLG0VE70</td>
<td>Data Attribute Inquiry Summary</td>
</tr>
<tr>
<td>BLG0VE71</td>
<td>Data Attribute Record Inquiry</td>
</tr>
<tr>
<td>BLG0VE72</td>
<td>Response Processing Inquiry</td>
</tr>
<tr>
<td>BLG0VE73</td>
<td>Structured Word and Validation Inquiry</td>
</tr>
<tr>
<td>BLG0VE74</td>
<td>Panel Flow Inquiry</td>
</tr>
<tr>
<td>BLG0VE75</td>
<td>Control Data Inquiry</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Version 1.1
<table>
<thead>
<tr>
<th>Panel ID</th>
<th>Description</th>
<th>Panel ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0V800</td>
<td>SRC Record Control</td>
<td>BLG00011</td>
<td>System Record Inquiry</td>
</tr>
<tr>
<td>BLG0V801</td>
<td>SRC Start Panel</td>
<td>BLG00020</td>
<td>Reference Entry</td>
</tr>
<tr>
<td>BLG0V802</td>
<td>SRC Application</td>
<td>BLG00030</td>
<td>Application Selection</td>
</tr>
<tr>
<td>BLG0W000</td>
<td>Report Entry</td>
<td>BLG00040</td>
<td>Data Model Entry</td>
</tr>
<tr>
<td>BLG0W010</td>
<td>General Reports</td>
<td>BLG00041</td>
<td>Data Model Inquiry</td>
</tr>
<tr>
<td>BLG0W020</td>
<td>Problem Reports</td>
<td>BLG00100</td>
<td>Response Type</td>
</tr>
<tr>
<td>BLG0W030</td>
<td>Change Reports</td>
<td>BLG00101</td>
<td>Verify Change Operation</td>
</tr>
<tr>
<td>BLG0W040</td>
<td>Configuration Reports</td>
<td>BLG00102</td>
<td>Verification</td>
</tr>
<tr>
<td>BLG0W200</td>
<td>Periodic Change Status Report</td>
<td>BLG00103</td>
<td>Delete Verification</td>
</tr>
<tr>
<td>BLG0W210</td>
<td>Periodic Problem Status Report</td>
<td>BLG00104</td>
<td>Verify Change Operation</td>
</tr>
<tr>
<td>BLG0W250</td>
<td>Extract Diagram Data Specification</td>
<td>BLG00105</td>
<td>Audit Trail Options</td>
</tr>
<tr>
<td>BLG0W510</td>
<td>Browse/Print</td>
<td>BLM0YE00</td>
<td>Primary Options Menu</td>
</tr>
<tr>
<td>BLG0X100</td>
<td>Glossary</td>
<td>BLM0YE50</td>
<td>Primary Options Menu</td>
</tr>
<tr>
<td>BLG0Y100</td>
<td>Database Administration Selection</td>
<td>BLM0YU00</td>
<td>Record Summary</td>
</tr>
<tr>
<td>BLG0Y200</td>
<td>Check In</td>
<td>BLM0YU10</td>
<td>Record Delete Verification</td>
</tr>
<tr>
<td>BLG0ZD50</td>
<td>Automatic Log Save Delete Verification</td>
<td>BLM0YU11</td>
<td>Record Delete Verification</td>
</tr>
<tr>
<td>BLG0ZD51</td>
<td>Automatic Log Save Delete Verification</td>
<td>BLM0YU50</td>
<td>Parent Record Summary</td>
</tr>
<tr>
<td>BLG0ZU00</td>
<td>Escalation Rules Summary</td>
<td>BLM0YU51</td>
<td>Parent Record Summary</td>
</tr>
<tr>
<td>BLG0ZU50</td>
<td>Automatic Log Save Record Summary</td>
<td>BLM0YU60</td>
<td>Record Delete Verification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0YU61</td>
<td>Record Delete Verification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0YU80</td>
<td>Child Record Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0YU90</td>
<td>Child Delete Verification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0Y91</td>
<td>Child Delete Verification</td>
</tr>
<tr>
<td>BLG0Z001</td>
<td>Rules Entry</td>
<td>BLM0Y100</td>
<td>Record Entry</td>
</tr>
<tr>
<td>BLG0Z020</td>
<td>Rules Text Entry</td>
<td>BLM0Y101</td>
<td>Panel Set Entry</td>
</tr>
<tr>
<td>BLG0Z100</td>
<td>Escalation Rules Entry</td>
<td>BLM0Y120</td>
<td>Text Entry</td>
</tr>
<tr>
<td>BLG0Z500</td>
<td>Automatic Log Save Data Set Information Entry</td>
<td>BLM0Y200</td>
<td>Record Description</td>
</tr>
<tr>
<td>BLG0Z510</td>
<td>Automatic Log Save Synchronize and Filter Entry</td>
<td>BLM0Y201</td>
<td>Record Description</td>
</tr>
<tr>
<td>BLG0Z530</td>
<td>DB2 Extract Information Entry</td>
<td>BLM0Y202</td>
<td>Record Description</td>
</tr>
<tr>
<td>BLG00000</td>
<td>INFO/Management Entry</td>
<td>BLM0Y203</td>
<td>Record Inquiry</td>
</tr>
<tr>
<td>BLG00001</td>
<td>INFO/Management Inquiry</td>
<td>BLM0Y250</td>
<td>Panel Set Inquiry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0Y251</td>
<td>Panel Set Inquiry Summary</td>
</tr>
<tr>
<td>Panel ID</td>
<td>Description</td>
<td>Panel ID</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------</td>
<td>---------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>BLM0Y260</td>
<td>Record Control</td>
<td>BLM0Y793</td>
<td>Record+ Inquiry</td>
</tr>
<tr>
<td>BLM0Y261</td>
<td>Record Control Data Inquiry</td>
<td>BLM0Y800</td>
<td>Record Summary Display</td>
</tr>
<tr>
<td>BLM0Y300</td>
<td>Record Summary Display</td>
<td>BLM0Y810</td>
<td>Parent Record Display</td>
</tr>
<tr>
<td>BLM0Y310</td>
<td>Panel Set Display</td>
<td>BLM0Y820</td>
<td>Display Record Text</td>
</tr>
<tr>
<td>BLM0Y320</td>
<td>Display Record Text</td>
<td>BLM0Y830</td>
<td>Record Utilities</td>
</tr>
<tr>
<td>BLM0Y330</td>
<td>Record Utilities</td>
<td>BLM0Y850</td>
<td>Child Summary Display</td>
</tr>
<tr>
<td>BLM0Y600</td>
<td>Parent Record Entry</td>
<td>BLM0Y860</td>
<td>Child Record Display</td>
</tr>
<tr>
<td>BLM0Y601</td>
<td>Record Entry</td>
<td>BLM0Y870</td>
<td>Display Child Record Text</td>
</tr>
<tr>
<td>BLM0Y620</td>
<td>Text Entry</td>
<td>BLM0Y880</td>
<td>Record Utilities</td>
</tr>
<tr>
<td>BLM0Y630</td>
<td>Text Entry</td>
<td>BTN0BU10</td>
<td>Problem Delete Verification</td>
</tr>
<tr>
<td>BLM0Y650</td>
<td>Child Record Entry</td>
<td>BTN0BU11</td>
<td>Problem Delete Verification</td>
</tr>
<tr>
<td>BLM0Y700</td>
<td>Parent Record Description</td>
<td>BTN0B001</td>
<td>Problem Entry</td>
</tr>
<tr>
<td>BLM0Y701</td>
<td>Parent Record Description</td>
<td>BTN0B002</td>
<td>Problem Update</td>
</tr>
<tr>
<td>BLM0Y702</td>
<td>Parent Record Inquiry</td>
<td>BTN0B010</td>
<td>Problem Text Entry</td>
</tr>
<tr>
<td>BLM0Y703</td>
<td>Parent Record Inquiry</td>
<td>BTN0B101</td>
<td>Problem Caller Information</td>
</tr>
<tr>
<td>BLM0Y704</td>
<td>Record Entry</td>
<td>BTN0B102</td>
<td>Initial Problem Data</td>
</tr>
<tr>
<td>BLM0Y710</td>
<td>Child Record Description</td>
<td>BTN0B200</td>
<td>Problem Status Entry</td>
</tr>
<tr>
<td>BLM0Y711</td>
<td>Child Record Description</td>
<td>BTN0B300</td>
<td>Problem Close Entry</td>
</tr>
<tr>
<td>BLM0Y712</td>
<td>Child Record Description</td>
<td>BTN0B500</td>
<td>Problem Synopsis Entry</td>
</tr>
<tr>
<td>BLM0Y713</td>
<td>Child Record Inquiry</td>
<td>BTN0CETX</td>
<td>Change Text Entry Panel</td>
</tr>
<tr>
<td>BLM0Y740</td>
<td>Parent/Child Inquiry</td>
<td>BTN0CU00</td>
<td>Change Request Summary</td>
</tr>
<tr>
<td>BLM0Y750</td>
<td>Record Inquiry</td>
<td>BTN0CU01</td>
<td>Change Requester Summary</td>
</tr>
<tr>
<td>BLM0Y751</td>
<td>Parent Inquiry Summary</td>
<td>BTN0C100</td>
<td>Change Requester Entry</td>
</tr>
<tr>
<td>BLM0Y760</td>
<td>Record Control</td>
<td>BTN0C101</td>
<td>Initial Change Data</td>
</tr>
<tr>
<td>BLM0Y761</td>
<td>Record Control Data Inquiry</td>
<td>BTN0C200</td>
<td>Change Status Entry</td>
</tr>
<tr>
<td>BLM0Y770</td>
<td>Child Record Inquiry</td>
<td>BTN0C300</td>
<td>Change Close Entry</td>
</tr>
<tr>
<td>BLM0Y771</td>
<td>Child Inquiry Summary</td>
<td>BTN0DU01</td>
<td>Hardware Component Summary</td>
</tr>
<tr>
<td>BLM0Y780</td>
<td>Parent/Child Inquiry</td>
<td>BTN0DU03</td>
<td>Software Component Summary</td>
</tr>
<tr>
<td>BLM0Y781</td>
<td>Parent/Child Inquiry Summary</td>
<td>BTN0DU08</td>
<td>Person Record</td>
</tr>
<tr>
<td>BLM0Y790</td>
<td>Record+ Inquiry</td>
<td>BTN0DU18</td>
<td>Person Delete Verification</td>
</tr>
<tr>
<td>BLM0Y791</td>
<td>Record+ Inquiry</td>
<td>BTN0DU28</td>
<td>Person Delete Verification</td>
</tr>
<tr>
<td>BLM0Y792</td>
<td>Record+ Inquiry</td>
<td>BTN0DU30</td>
<td>Hardware Component Summary</td>
</tr>
</tbody>
</table>
Reference

Information/Management Panels Arranged by Panel ID

BTN0DU31  Software Component Summary
BTN0D000  Configuration Entry
BTN0D100  Hardware Component Entry
BTN0D101  Component Entry
BTN0D102  Hardware Component Entry
BTN0D104  Hardware Component Entry
BTN0D200  Software Component Entry
BTN0D201  Software Component Entry
BTN0D202  Software Component Entry
BTN0D500  Person Record
BTN0EA90  Search List Options
BTN0EA91  Search List Options
BTN0ENCC  Change Controller Primary Options
BTN0ENCN  Configuration Controller Primary Options
BTN0ENHD  Help Desk Primary Options
BTN0ENNO  Network Support Primary Options
BTN0ENOM  Operations Manager Primary Options
BTN0ENPC  Problem Controller Primary Options
BTN0ENSU  Support Primary Options
BTN0ENSY  System Administrator Primary Options
BTN0EPPC  Privilege Class Selection
BTN0E000  Quick Problem Search Panel
BTN0E090  Problem Inquiry Summary
BTN0E190  Problem Reporter Data Inquiry
BTN0E290  Problem Status Data Inquiry
BTN0E390  Problem Close Data Inquiry
BTN0E890  Control Data Inquiry
BTN0F000  Quick Change Search Panel
BTN0F090  Change Request Summary
BTN0F190  Change Requester Inquiry
BTN0F290  Change Status Data Inquiry
BTN0F390  Change Close Data Inquiry
BTN0G000  Quick Config Search Panel
BTN0G001  Hardware Component Update
BTN0G002  Software Component Update
BTN0G1A1  Software Component Inquiry
BTN0G1A3  Software Component Inquiry Summary
BTN0G190  Hardware Component Inquiry Summary
BTN0G19A  Search List Options
BTN0H000  Person Search Option
BTN0H001  Person Search Option
BTN0J100  Class Description Entry
BTN0J200  Authority Entry
BTN0J300  Class User Entry
BTN0L010  Display Problem Text
BTN0L100  Problem Reporter Display
BTN0L200  Problem Status Display
BTN0L300  Problem Close Display
BTN0L500  Problem Synopsis Display
BTN0M010  Change Text Display
BTN0M100  Change Requester Display
BTN0M200  Change Status Display
BTN0N100  Hardware Component Summary Display
BTN0N101  Hardware Component Display
BTN0N131  Connectivity Display Options
BTN0N134  Path Display Specification
<table>
<thead>
<tr>
<th>Panel ID</th>
<th>Panel Name</th>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTN0N2000</td>
<td>Software Component Summary Display</td>
<td>BTN0XDES</td>
<td>Class Description Entry</td>
</tr>
<tr>
<td>BTN0N2001</td>
<td>Software Component Display</td>
<td>BTN0XHID</td>
<td>Class Description Entry</td>
</tr>
<tr>
<td>BTN0N500</td>
<td>Person Record</td>
<td>BTN0XLPC</td>
<td>Load Privilege Class Record</td>
</tr>
<tr>
<td>BTN0N510</td>
<td>Person Record</td>
<td>BTN0XSUM</td>
<td>Class Summary</td>
</tr>
<tr>
<td>BTN0Q100</td>
<td>Class Summary Display</td>
<td>BTN0XUSR</td>
<td>Class User Entry</td>
</tr>
<tr>
<td>BTN0Q200</td>
<td>Authority Display</td>
<td>BTN00000</td>
<td>INFO/Management Entry</td>
</tr>
<tr>
<td>BTN0Q300</td>
<td>Class User Entry</td>
<td>BTN00001</td>
<td>INFO/Management Inquiry</td>
</tr>
<tr>
<td>BTN0S010</td>
<td>Problem Summary Display</td>
<td>EYM5M100</td>
<td>Mapping Reference Entry</td>
</tr>
<tr>
<td>BTN0S020</td>
<td>Change Summary Display</td>
<td>EYM5M500</td>
<td>Mapping Reference Data Entry</td>
</tr>
<tr>
<td>BTN0S021</td>
<td>Record Utilities</td>
<td>EYMA100</td>
<td>Mapping Reference Inquiry</td>
</tr>
<tr>
<td>BTN0XAUT</td>
<td>Authority Entry</td>
<td>EYMBM100</td>
<td>Mapping Reference Display</td>
</tr>
<tr>
<td>BTN0XAUT</td>
<td>Authority Entry</td>
<td>EYMBM500</td>
<td>Mapping Reference Data Display</td>
</tr>
</tbody>
</table>

**Information/Management Panels Arranged by Panel Title**

<table>
<thead>
<tr>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0F230</td>
<td>Activity Close</td>
</tr>
<tr>
<td>BLG0F231</td>
<td>Activity Close</td>
</tr>
<tr>
<td>BLG0F283</td>
<td>Activity Close Data Inquiry</td>
</tr>
<tr>
<td>BLG0M630</td>
<td>Activity Close Display</td>
</tr>
<tr>
<td>BLG0C630</td>
<td>Activity Close Entry</td>
</tr>
<tr>
<td>BLG0F171</td>
<td>Activity Control</td>
</tr>
<tr>
<td>BLG0F174</td>
<td>Activity Control</td>
</tr>
<tr>
<td>BLG0F288</td>
<td>Activity Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0CU30</td>
<td>Activity Delete Verification</td>
</tr>
<tr>
<td>BLG0CU31</td>
<td>Activity Delete Verification</td>
</tr>
<tr>
<td>BLG0C640</td>
<td>Activity Detail</td>
</tr>
<tr>
<td>BLG0C003</td>
<td>Activity Entry</td>
</tr>
<tr>
<td>BLG0F200</td>
<td>Activity Inquiry</td>
</tr>
<tr>
<td>BLG0F210</td>
<td>Activity Requester</td>
</tr>
<tr>
<td>BLG0F211</td>
<td>Activity Requester</td>
</tr>
<tr>
<td>BLG0F281</td>
<td>Activity Requester Data Inquiry</td>
</tr>
<tr>
<td>BLG0M600</td>
<td>Activity Requester Display</td>
</tr>
<tr>
<td>BLG0C600</td>
<td>Activity Requester Entry</td>
</tr>
<tr>
<td>EYM5M100</td>
<td>Mapping Reference Entry</td>
</tr>
<tr>
<td>EYM5M500</td>
<td>Mapping Reference Data Entry</td>
</tr>
<tr>
<td>EYMA100</td>
<td>Mapping Reference Inquiry</td>
</tr>
<tr>
<td>EYMBM100</td>
<td>Mapping Reference Display</td>
</tr>
<tr>
<td>EYMBM500</td>
<td>Mapping Reference Data Display</td>
</tr>
</tbody>
</table>

**Information/Management Panels Arranged by Panel ID**

<table>
<thead>
<tr>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTN0N2000</td>
<td>Software Component Summary Display</td>
</tr>
<tr>
<td>BTN0N2001</td>
<td>Software Component Display</td>
</tr>
<tr>
<td>BTN0N500</td>
<td>Person Record</td>
</tr>
<tr>
<td>BTN0N510</td>
<td>Person Record</td>
</tr>
<tr>
<td>BTN0Q100</td>
<td>Class Summary Display</td>
</tr>
<tr>
<td>BTN0Q200</td>
<td>Authority Display</td>
</tr>
<tr>
<td>BTN0Q300</td>
<td>Class User Entry</td>
</tr>
<tr>
<td>BTN0S010</td>
<td>Problem Summary Display</td>
</tr>
<tr>
<td>BTN0S020</td>
<td>Change Summary Display</td>
</tr>
<tr>
<td>BTN0S021</td>
<td>Record Utilities</td>
</tr>
<tr>
<td>BTN0XAUT</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BTN00000</td>
<td>INFO/Management Entry</td>
</tr>
<tr>
<td>BTN00001</td>
<td>INFO/Management Inquiry</td>
</tr>
<tr>
<td>EYM5M100</td>
<td>Mapping Reference Entry</td>
</tr>
<tr>
<td>EYM5M500</td>
<td>Mapping Reference Data Entry</td>
</tr>
<tr>
<td>EYMA100</td>
<td>Mapping Reference Inquiry</td>
</tr>
<tr>
<td>EYMBM100</td>
<td>Mapping Reference Display</td>
</tr>
<tr>
<td>EYMBM500</td>
<td>Mapping Reference Data Display</td>
</tr>
</tbody>
</table>

**Information/Management Panels Arranged by Panel Title**

<table>
<thead>
<tr>
<th>Panel ID</th>
<th>Panel Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0F230</td>
<td>Activity Close</td>
</tr>
<tr>
<td>BLG0F231</td>
<td>Activity Close</td>
</tr>
<tr>
<td>BLG0F283</td>
<td>Activity Close Data Inquiry</td>
</tr>
<tr>
<td>BLG0M630</td>
<td>Activity Close Display</td>
</tr>
<tr>
<td>BLG0C630</td>
<td>Activity Close Entry</td>
</tr>
<tr>
<td>BLG0F171</td>
<td>Activity Control</td>
</tr>
<tr>
<td>BLG0F174</td>
<td>Activity Control</td>
</tr>
<tr>
<td>BLG0F288</td>
<td>Activity Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0CU30</td>
<td>Activity Delete Verification</td>
</tr>
<tr>
<td>BLG0CU31</td>
<td>Activity Delete Verification</td>
</tr>
<tr>
<td>BLG0C640</td>
<td>Activity Detail</td>
</tr>
<tr>
<td>BLG0C003</td>
<td>Activity Entry</td>
</tr>
<tr>
<td>BLG0F200</td>
<td>Activity Inquiry</td>
</tr>
<tr>
<td>BLG0F210</td>
<td>Activity Requester</td>
</tr>
<tr>
<td>BLG0F211</td>
<td>Activity Requester</td>
</tr>
<tr>
<td>BLG0F281</td>
<td>Activity Requester Data Inquiry</td>
</tr>
<tr>
<td>BLG0M600</td>
<td>Activity Requester Display</td>
</tr>
<tr>
<td>BLG0C600</td>
<td>Activity Requester Entry</td>
</tr>
<tr>
<td>EYM5M100</td>
<td>Mapping Reference Entry</td>
</tr>
<tr>
<td>EYM5M500</td>
<td>Mapping Reference Data Entry</td>
</tr>
<tr>
<td>EYMA100</td>
<td>Mapping Reference Inquiry</td>
</tr>
<tr>
<td>EYMBM100</td>
<td>Mapping Reference Display</td>
</tr>
<tr>
<td>EYMBM500</td>
<td>Mapping Reference Data Display</td>
</tr>
<tr>
<td>Reference</td>
<td>Information/Management Panels Arranged by Panel Title</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>BLG0F440</td>
<td>All Changes</td>
</tr>
<tr>
<td>BLG0F445</td>
<td>All Changes</td>
</tr>
<tr>
<td>BLG0G337</td>
<td>All Financial+ Inquiry</td>
</tr>
<tr>
<td>BLG0G338</td>
<td>All Financial+ Summary</td>
</tr>
<tr>
<td>BLG010E0</td>
<td>APAR Display</td>
</tr>
<tr>
<td>BLG00030</td>
<td>Application Selection</td>
</tr>
<tr>
<td>BLG0F150</td>
<td>Approver Classes</td>
</tr>
<tr>
<td>BLG1TARG</td>
<td>Argument -- Review or Modify Current Search Argument</td>
</tr>
<tr>
<td>BLG600AG</td>
<td>Assignee Department</td>
</tr>
<tr>
<td>BTN600AG</td>
<td>Assignee Department</td>
</tr>
<tr>
<td>BLG600AA</td>
<td>Assignee Name</td>
</tr>
<tr>
<td>BTN600C1</td>
<td>Assignee Name</td>
</tr>
<tr>
<td>BTN600PA</td>
<td>Assignee Name or Class</td>
</tr>
<tr>
<td>BLG00105</td>
<td>Audit Trail Options</td>
</tr>
<tr>
<td>BLG0Q210</td>
<td>Authority Display</td>
</tr>
<tr>
<td>BLG0Q211</td>
<td>Authority Display</td>
</tr>
<tr>
<td>BLG0Q212</td>
<td>Authority Display</td>
</tr>
<tr>
<td>BTN0Q200</td>
<td>Authority Display</td>
</tr>
<tr>
<td>BLG0J210</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BLG0J211</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BLG0J212</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BTN0J200</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BTN0XAUT</td>
<td>Authority Entry</td>
</tr>
<tr>
<td>BLG0K210</td>
<td>Authority Inquiry</td>
</tr>
<tr>
<td>BLG0K211</td>
<td>Authority Inquiry</td>
</tr>
<tr>
<td>BLG0K212</td>
<td>Authority Inquiry</td>
</tr>
<tr>
<td>BLG0Q213</td>
<td>Authority Summary</td>
</tr>
<tr>
<td>BLG0Q214</td>
<td>Authority Summary</td>
</tr>
<tr>
<td>BLG0Q215</td>
<td>Authority Summary</td>
</tr>
<tr>
<td>BLG0Q216</td>
<td>Authority Summary</td>
</tr>
<tr>
<td>BLG0VQ62</td>
<td>Authorization Code Display</td>
</tr>
<tr>
<td>BLG0V610</td>
<td>Authorization Code Entry</td>
</tr>
<tr>
<td>BLG0VE62</td>
<td>Authorization Code Inquiry</td>
</tr>
<tr>
<td>BLGLQSPL</td>
<td>Authorized Logical Partition Identifier Display</td>
</tr>
<tr>
<td>BLGLJSPL</td>
<td>Authorized Logical Partition Identifier Entry</td>
</tr>
<tr>
<td>BLG0S500</td>
<td>Automatic Log Save Data Set Information Display</td>
</tr>
<tr>
<td>BLG0Z500</td>
<td>Automatic Log Save Data Set Information Entry</td>
</tr>
<tr>
<td>BLG0ZD50</td>
<td>Automatic Log Save Delete Verification</td>
</tr>
<tr>
<td>BLG0ZD51</td>
<td>Automatic Log Save Delete Verification</td>
</tr>
<tr>
<td>BLG0Z510</td>
<td>Automatic Log Save Synchronize and Filter Display</td>
</tr>
<tr>
<td>BLG0Z510</td>
<td>Automatic Log Save Synchronize and Filter Entry</td>
</tr>
<tr>
<td>BLG0ZU50</td>
<td>Automatic Log Save Record Summary</td>
</tr>
<tr>
<td>BLG0C495</td>
<td>Bill of Materials</td>
</tr>
<tr>
<td>BLG0F495</td>
<td>Bill of Materials</td>
</tr>
<tr>
<td>BLG0W510</td>
<td>Browse/Print</td>
</tr>
<tr>
<td>BTN6URN2</td>
<td>Caller's Identifier</td>
</tr>
<tr>
<td>BTN6URN4</td>
<td>Caller's Identifier</td>
</tr>
<tr>
<td>BTN7URN4</td>
<td>Caller's Identifier</td>
</tr>
<tr>
<td>BLG0DU18</td>
<td>Center Delete Verification</td>
</tr>
<tr>
<td>BLG0DU28</td>
<td>Center Delete Verification</td>
</tr>
<tr>
<td>BLG0G511</td>
<td>Center Inquiry</td>
</tr>
<tr>
<td>BLG0DU71</td>
<td>Center Rename Verification</td>
</tr>
<tr>
<td>BLG0F590</td>
<td>Change Approver Data Inquiry</td>
</tr>
</tbody>
</table>
Information/Management Panels Arranged by Panel Title

BLG0M500 Change Approver Display
BLG0C500 Change Approver Entry
BLG0J220 Change Authority
BLG0K220 Change Authority
BLG0Q220 Change Authority
BLG0F130 Change Close
BLG0F131 Change Close
BLG0F390 Change Close Data Inquiry
BTN0F390 Change Close Data Inquiry
BLG0M300 Change Close Display
BLG0C300 Change Close Entry
BTN0C300 Change Close Entry
BLG0F172 Change Control
BLG0F175 Change Control
BLG0F890 Change Control Data Inquiry
BTN0ENCC Change Controller Primary Options
BLG0CU10 Change Delete Verification
BLG0CU11 Change Delete Verification
BLG0C040 Change Detail
BLG0C400 Change Detail
BLG0F400 Change Detail
BLG0F450 Change Detail
BLG0C004 Change Entry
BLG0F100 Change Inquiry
BLG0W030 Change Reports
BLG0CU00 Change Request Summary
BLG0CU01 Change Request Summary
BLG0F090 Change Request Summary
BTN0CU00 Change Request Summary
BTN0CU01 Change Request Summary
BTN0F090 Change Request Summary
BLG0F110 Change Requester
BLG0F111 Change Requester
BLG0M100 Change Requester Display
BTN0M100 Change Requester Display
BLG0C100 Change Requester Entry
BTN0C100 Change Requester Entry
BLG0F190 Change Requester Inquiry
BTN0F190 Change Requester Inquiry
BLG0M700 Change Reviewer Display
BLG0C700 Change Reviewer Entry
BLG0F120 Change Status
BLG0F121 Change Status
BLG0F122 Change Status
BLG0F290 Change Status Data Inquiry
BTN0F290 Change Status Data Inquiry
BLG0M200 Change Status Display
BTN0M200 Change Status Display
BLG0C200 Change Status Entry
BTN0C200 Change Status Entry
BLG0S020 Change Summary Display
BTN0S020 Change Summary Display
BLG0C010 Change Text Display
BTN0CETX Change Text Entry Panel
BLG0F330 Change/Activity Close
BLG0F331 Change/Activity Close
BLG0F273 Change/Activity Close Data Inquiry
BLG0F170 Change/Activity Control
BLG0F173 Change/Activity Control
BLG0F278 Change/Activity Control Data Inquiry
<table>
<thead>
<tr>
<th>Reference</th>
<th>Information/Management Panels Arranged by Panel Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0F000</td>
<td>Change/Activity Inquiry</td>
</tr>
<tr>
<td>BLG0F300</td>
<td>Change/Activity Inquiry</td>
</tr>
<tr>
<td>BLG6CRES</td>
<td>Change/Activity Reason</td>
</tr>
<tr>
<td>BLG0F270</td>
<td>Change/Activity Request Summary</td>
</tr>
<tr>
<td>BLG0F310</td>
<td>Change/Activity Requester</td>
</tr>
<tr>
<td>BLG0F311</td>
<td>Change/Activity Requester</td>
</tr>
<tr>
<td>BLG0F271</td>
<td>Change/Activity Requester Data Inquiry</td>
</tr>
<tr>
<td>BLG0F320</td>
<td>Change/Activity Status</td>
</tr>
<tr>
<td>BLG0F321</td>
<td>Change/Activity Status</td>
</tr>
<tr>
<td>BLG0F322</td>
<td>Change/Activity Status</td>
</tr>
<tr>
<td>BLG0F272</td>
<td>Change/Activity Status Data Inquiry</td>
</tr>
<tr>
<td>BLG0Y200</td>
<td>Check In</td>
</tr>
<tr>
<td>BLM0YU90</td>
<td>Child Delete Verification</td>
</tr>
<tr>
<td>BLM0YU91</td>
<td>Child Delete Verification</td>
</tr>
<tr>
<td>BLM0Y771</td>
<td>Child Inquiry Summary</td>
</tr>
<tr>
<td>BLM0Y710</td>
<td>Child Record Description</td>
</tr>
<tr>
<td>BLM0Y711</td>
<td>Child Record Description</td>
</tr>
<tr>
<td>BLM0Y712</td>
<td>Child Record Description</td>
</tr>
<tr>
<td>BLM0Y860</td>
<td>Child Record Display</td>
</tr>
<tr>
<td>BLM0Y650</td>
<td>Child Record Entry</td>
</tr>
<tr>
<td>BLM0Y713</td>
<td>Child Record Inquiry</td>
</tr>
<tr>
<td>BLM0Y770</td>
<td>Child Record Inquiry</td>
</tr>
<tr>
<td>BLM1YCCH</td>
<td>Child Record List</td>
</tr>
<tr>
<td>BLM0YU80</td>
<td>Child Record Summary</td>
</tr>
<tr>
<td>BLM0Y850</td>
<td>Child Summary Display</td>
</tr>
<tr>
<td>BLG0J260</td>
<td>Class Authority</td>
</tr>
<tr>
<td>BLG0K260</td>
<td>Class Authority</td>
</tr>
<tr>
<td>BLG0Q260</td>
<td>Class Authority</td>
</tr>
<tr>
<td>BLG0JU10</td>
<td>Class Delete Verification</td>
</tr>
<tr>
<td>BLG0JU11</td>
<td>Class Delete Verification</td>
</tr>
<tr>
<td>BLG0K100</td>
<td>Class Description</td>
</tr>
<tr>
<td>BLG0J100</td>
<td>Class Description Entry</td>
</tr>
<tr>
<td>BTN0J100</td>
<td>Class Description Entry</td>
</tr>
<tr>
<td>BTN0XDES</td>
<td>Class Description Entry</td>
</tr>
<tr>
<td>BTN0XHID</td>
<td>Class Description Entry</td>
</tr>
<tr>
<td>BLG0K190</td>
<td>Class Description Inquiry</td>
</tr>
<tr>
<td>BLG0K000</td>
<td>Class Inquiry</td>
</tr>
<tr>
<td>BTN6PNAM</td>
<td>Class Name</td>
</tr>
<tr>
<td>BLG0JU00</td>
<td>Class Summary</td>
</tr>
<tr>
<td>BLG0K091</td>
<td>Class Summary</td>
</tr>
<tr>
<td>BTN0JU00</td>
<td>Class Summary</td>
</tr>
<tr>
<td>BTN0XSUM</td>
<td>Class Summary</td>
</tr>
<tr>
<td>BLG0Q100</td>
<td>Class Summary Display</td>
</tr>
<tr>
<td>BTN0Q100</td>
<td>Class Summary Display</td>
</tr>
<tr>
<td>BLG0J300</td>
<td>Class User Entry</td>
</tr>
<tr>
<td>BTN0J300</td>
<td>Class User Entry</td>
</tr>
<tr>
<td>BTN0Q300</td>
<td>Class User Entry</td>
</tr>
<tr>
<td>BTN0XUSR</td>
<td>Class User Entry</td>
</tr>
<tr>
<td>BLGLCMDD</td>
<td>Command Alias/Authorization Display</td>
</tr>
<tr>
<td>BLGLCMDU</td>
<td>Command Alias/Authorization Entry</td>
</tr>
<tr>
<td>BLG0D101</td>
<td>Component Entry</td>
</tr>
<tr>
<td>BLG0D109</td>
<td>Component Entry</td>
</tr>
<tr>
<td>BLG0D110</td>
<td>Component Entry</td>
</tr>
<tr>
<td>BTN0D101</td>
<td>Component Entry</td>
</tr>
<tr>
<td>BLG0G100</td>
<td>Component Records</td>
</tr>
<tr>
<td>BLG0G101</td>
<td>Component Records</td>
</tr>
<tr>
<td>BLG0G186</td>
<td>Component Records+</td>
</tr>
<tr>
<td>BLG0G196</td>
<td>Component Records+</td>
</tr>
<tr>
<td>BLG0G1D0</td>
<td>Component Secondary Support</td>
</tr>
<tr>
<td>BLG0G1A0</td>
<td>Component Support</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>BLG0G185</td>
<td>Component+ Inquiry</td>
</tr>
<tr>
<td>BLG0J230</td>
<td>Configuration Authority</td>
</tr>
<tr>
<td>BLG0K230</td>
<td>Configuration Authority</td>
</tr>
<tr>
<td>BLG0Q230</td>
<td>Configuration Authority</td>
</tr>
<tr>
<td>BTN0ENCF</td>
<td>Configuration Controller Primary</td>
</tr>
<tr>
<td></td>
<td>Options</td>
</tr>
<tr>
<td>BLG0D000</td>
<td>Configuration Entry</td>
</tr>
<tr>
<td>BLG0G000</td>
<td>Configuration Inquiry</td>
</tr>
<tr>
<td>BLG0W040</td>
<td>Configuration Reports</td>
</tr>
<tr>
<td>BLG0G126</td>
<td>Connection Inquiry</td>
</tr>
<tr>
<td>BLG0G160</td>
<td>Connection Inquiry</td>
</tr>
<tr>
<td>BLG0G181</td>
<td>Connection Records+</td>
</tr>
<tr>
<td>BLG0E890</td>
<td>Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0VE75</td>
<td>Control Data Inquiry</td>
</tr>
<tr>
<td>BTN0E890</td>
<td>Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0G890</td>
<td>Control Inquiry</td>
</tr>
<tr>
<td>BLG0C493</td>
<td>Control Item</td>
</tr>
<tr>
<td>BLG0F493</td>
<td>Control Item</td>
</tr>
<tr>
<td>BLG0G893</td>
<td>Control Summary</td>
</tr>
<tr>
<td>BLG0D500</td>
<td>Data Center entry</td>
</tr>
<tr>
<td>BLG0VE70</td>
<td>Data Attribute Inquiry Summary</td>
</tr>
<tr>
<td>BLG0VD70</td>
<td>Data Attribute Record Delete</td>
</tr>
<tr>
<td></td>
<td>Verification</td>
</tr>
<tr>
<td>BLG0VD71</td>
<td>Data Attribute Record Delete</td>
</tr>
<tr>
<td></td>
<td>Verification</td>
</tr>
<tr>
<td>BLG0VQ71</td>
<td>Data Attribute Record Display</td>
</tr>
<tr>
<td>BLG0V700</td>
<td>Data Attribute Record Entry</td>
</tr>
<tr>
<td>BLG0V701</td>
<td>Data Attribute Record Entry</td>
</tr>
<tr>
<td>BLG0VE71</td>
<td>Data Attribute Record Inquiry</td>
</tr>
<tr>
<td>BLG0DATT</td>
<td>Data Attribute Records Display</td>
</tr>
<tr>
<td>BLGLDATT</td>
<td>Data Attribute Records Entry</td>
</tr>
<tr>
<td>Panel Code</td>
<td>Panel Title</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>BLG0N025</td>
<td>Display Connection Text</td>
</tr>
<tr>
<td>BLG0N080</td>
<td>Display Data Center Text</td>
</tr>
<tr>
<td>BLG0N020</td>
<td>Display Feature Text</td>
</tr>
<tr>
<td>BLG0N050</td>
<td>Display Hardware Financial Text</td>
</tr>
<tr>
<td>BLG0N010</td>
<td>Display Hardware Text</td>
</tr>
<tr>
<td>BLG0Q010</td>
<td>Display Privilege Class Text</td>
</tr>
<tr>
<td>BLG0L010</td>
<td>Display Problem Text</td>
</tr>
<tr>
<td>BTN0L010</td>
<td>Display Problem Text</td>
</tr>
<tr>
<td>BLM0Y320</td>
<td>Display Record Text</td>
</tr>
<tr>
<td>BLM0Y820</td>
<td>Display Record Text</td>
</tr>
<tr>
<td>BLG0R030</td>
<td>Display Rules Text</td>
</tr>
<tr>
<td>BLG0N060</td>
<td>Display Service Text</td>
</tr>
<tr>
<td>BLG0N095</td>
<td>Display Software Connection Text</td>
</tr>
<tr>
<td>BLG0N090</td>
<td>Display Software Feature Text</td>
</tr>
<tr>
<td>BLG0N051</td>
<td>Display Software Financial Text</td>
</tr>
<tr>
<td>BLG0N030</td>
<td>Display Software Text</td>
</tr>
<tr>
<td>BLG0T130</td>
<td>Display SRC Responses</td>
</tr>
<tr>
<td>BLG0N040</td>
<td>Display Subcomponent Text</td>
</tr>
<tr>
<td>BLG0N070</td>
<td>Display System Text</td>
</tr>
<tr>
<td>BLG0C071</td>
<td>Document Add</td>
</tr>
<tr>
<td>BLG0C171</td>
<td>Document Add Inquiry</td>
</tr>
<tr>
<td>BLG0C072</td>
<td>Document Alter</td>
</tr>
<tr>
<td>BLG0C172</td>
<td>Document Alter Inquiry</td>
</tr>
<tr>
<td>BLG0C070</td>
<td>Document Change</td>
</tr>
<tr>
<td>BLG0C073</td>
<td>Document Remove</td>
</tr>
<tr>
<td>BLG0C173</td>
<td>Document Remove Inquiry</td>
</tr>
<tr>
<td>BLG0C430</td>
<td>Documentation Change</td>
</tr>
<tr>
<td>BLG0F430</td>
<td>Documentation Change</td>
</tr>
<tr>
<td>BLG0F431</td>
<td>Documentation Change</td>
</tr>
<tr>
<td>BLG0P820</td>
<td>Draw Report Data Set Destination Entry</td>
</tr>
<tr>
<td>Panel Code</td>
<td>Panel Title</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>BLG0G125</td>
<td>Feature Inquiry</td>
</tr>
<tr>
<td>BLG0G155</td>
<td>Feature Inquiry</td>
</tr>
<tr>
<td>BLGLDFEA</td>
<td>Feature Name Entry</td>
</tr>
<tr>
<td>BLG0G176</td>
<td>Feature Records+</td>
</tr>
<tr>
<td>BLG0G175</td>
<td>Feature+ Inquiry</td>
</tr>
<tr>
<td>BLG0T120</td>
<td>Final Responses Display</td>
</tr>
<tr>
<td>BLG0U120</td>
<td>Final SRC Response Entry</td>
</tr>
<tr>
<td>BLG0J240</td>
<td>Financial Authority</td>
</tr>
<tr>
<td>BLG0K240</td>
<td>Financial Authority</td>
</tr>
<tr>
<td>BLG0Q240</td>
<td>Financial Authority</td>
</tr>
<tr>
<td>BLG0D320</td>
<td>Financial Entry</td>
</tr>
<tr>
<td>BLG0D351</td>
<td>Financial Entry</td>
</tr>
<tr>
<td>BLG0D352</td>
<td>Financial Entry</td>
</tr>
<tr>
<td>BLG0D354</td>
<td>Financial Entry from Record Copy</td>
</tr>
<tr>
<td>BLG0G1B0</td>
<td>Financial Information</td>
</tr>
<tr>
<td>BLG0G300</td>
<td>Financial Inquiry</td>
</tr>
<tr>
<td>BLG0G336</td>
<td>Financial Records+</td>
</tr>
<tr>
<td>BLG0G335</td>
<td>Financial+ Inquiry</td>
</tr>
<tr>
<td>BLG0W010</td>
<td>General Reports</td>
</tr>
<tr>
<td>BLG0F423</td>
<td>Generic Hardware</td>
</tr>
<tr>
<td>BLG0X100</td>
<td>Glossary</td>
</tr>
<tr>
<td>BLG0C061</td>
<td>Hardware Add</td>
</tr>
<tr>
<td>BLG0C161</td>
<td>Hardware Add Inquiry</td>
</tr>
<tr>
<td>BLG0C062</td>
<td>Hardware Alter</td>
</tr>
<tr>
<td>BLG0C063</td>
<td>Hardware Alter by Installing</td>
</tr>
<tr>
<td>BLG0C163</td>
<td>Hardware Alter by Installing Inquiry</td>
</tr>
<tr>
<td>BLG0C064</td>
<td>Hardware Alter by Modifying</td>
</tr>
<tr>
<td>BLG0C164</td>
<td>Hardware Alter by Modifying Inquiry</td>
</tr>
<tr>
<td>BLG0C065</td>
<td>Hardware Alter by Removing</td>
</tr>
</tbody>
</table>

A-26

Version 1.1
<table>
<thead>
<tr>
<th>Reference</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTN0G190</td>
<td>Hardware Component Inquiry Summary</td>
</tr>
<tr>
<td>BLG0N170</td>
<td>Hardware Component Line Display</td>
</tr>
<tr>
<td>BLG0D170</td>
<td>Hardware Component Line Entry</td>
</tr>
<tr>
<td>BLG0N172</td>
<td>Hardware Component Loop Display</td>
</tr>
<tr>
<td>BLG0D172</td>
<td>Hardware Component Loop Entry</td>
</tr>
<tr>
<td>BLG0N150</td>
<td>Hardware Component Primary Support Display</td>
</tr>
<tr>
<td>BLG0D150</td>
<td>Hardware Component Primary Support Entry</td>
</tr>
<tr>
<td>BLG0N110</td>
<td>Hardware Component Record Utilities</td>
</tr>
<tr>
<td>BLG0DU57</td>
<td>Hardware Component Rename Verification</td>
</tr>
<tr>
<td>BLG0N151</td>
<td>Hardware Component Secondary Support Display</td>
</tr>
<tr>
<td>BLG0D151</td>
<td>Hardware Component Secondary Support Entry</td>
</tr>
<tr>
<td>BLG0D105</td>
<td>Hardware Component Source Definition</td>
</tr>
<tr>
<td>BLG0N105</td>
<td>Hardware Component Source Display</td>
</tr>
<tr>
<td>BLG0DU01</td>
<td>Hardware Component Summary</td>
</tr>
<tr>
<td>BLG0DU30</td>
<td>Hardware Component Summary</td>
</tr>
<tr>
<td>BTN0DU01</td>
<td>Hardware Component Summary</td>
</tr>
<tr>
<td>BTN0DU30</td>
<td>Hardware Component Summary</td>
</tr>
<tr>
<td>BLG0N100</td>
<td>Hardware Component Summary Display</td>
</tr>
<tr>
<td>BTN0N100</td>
<td>Hardware Component Summary Display</td>
</tr>
<tr>
<td>BLG0DU74</td>
<td>Hardware Component Swap Verification</td>
</tr>
<tr>
<td>BTN0G001</td>
<td>Hardware Component Update</td>
</tr>
<tr>
<td>BLG0G110</td>
<td>Hardware Components</td>
</tr>
<tr>
<td>BLG0G115</td>
<td>Hardware Components</td>
</tr>
<tr>
<td>BLG0G113</td>
<td>Hardware Components (Secondary)</td>
</tr>
<tr>
<td>BLG0G127</td>
<td>Hardware Connection</td>
</tr>
<tr>
<td>BLG0G128</td>
<td>Hardware Connection</td>
</tr>
<tr>
<td>BLG0DU34</td>
<td>Hardware Connection Delete Verification</td>
</tr>
<tr>
<td>BLG0DU35</td>
<td>Hardware Connection Delete Verification</td>
</tr>
<tr>
<td>BLG0N130</td>
<td>Hardware Connection Display</td>
</tr>
<tr>
<td>BLG0N132</td>
<td>Hardware Connection Display</td>
</tr>
<tr>
<td>BLG0D130</td>
<td>Hardware Connection Entry</td>
</tr>
<tr>
<td>BLG0G1A4</td>
<td>Hardware Connection Inquiry</td>
</tr>
<tr>
<td>BLG0N133</td>
<td>Hardware Connection Record Utilities</td>
</tr>
<tr>
<td>BLG0DU58</td>
<td>Hardware Connection Rename Verification</td>
</tr>
<tr>
<td>BLG0DU32</td>
<td>Hardware Connection Summary</td>
</tr>
<tr>
<td>BLG0G1A5</td>
<td>Hardware Connection Summary</td>
</tr>
<tr>
<td>BLG0D106</td>
<td>Hardware Connectivity Entry</td>
</tr>
<tr>
<td>BLG0G120</td>
<td>Hardware Feature</td>
</tr>
<tr>
<td>BLG0G121</td>
<td>Hardware Feature</td>
</tr>
<tr>
<td>BLG0DU12</td>
<td>Hardware Feature Delete Verification</td>
</tr>
<tr>
<td>BLG0DU22</td>
<td>Hardware Feature Delete Verification</td>
</tr>
<tr>
<td>BLG0G184</td>
<td>Hardware Feature Description/Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0N122</td>
<td>Hardware Feature Display</td>
</tr>
<tr>
<td>BLG0D103</td>
<td>Hardware Feature Entry</td>
</tr>
<tr>
<td>BLG0D120</td>
<td>Hardware Feature Entry</td>
</tr>
<tr>
<td>BLG0N154</td>
<td>Hardware Feature Financial Display</td>
</tr>
<tr>
<td>BLG0D154</td>
<td>Hardware Feature Financial Entry</td>
</tr>
<tr>
<td>Panel Code</td>
<td>Panel Title</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td>BLG0N121</td>
<td>Hardware Feature Record Utilities</td>
</tr>
<tr>
<td>BLG0DU59</td>
<td>Hardware Feature Rename Verification</td>
</tr>
<tr>
<td>BLG0DU02</td>
<td>Hardware Feature Summary</td>
</tr>
<tr>
<td>BLG0G183</td>
<td>Hardware Feature Summary</td>
</tr>
<tr>
<td>BLG0N120</td>
<td>Hardware Feature Summary Display</td>
</tr>
<tr>
<td>BLG0N157</td>
<td>Hardware Feature Support Display</td>
</tr>
<tr>
<td>BLG0D157</td>
<td>Hardware Feature Support Entry</td>
</tr>
<tr>
<td>BLG0G182</td>
<td>Hardware Feature Support/EC Levels/Financial Data Inquiry</td>
</tr>
<tr>
<td>BLG0G310</td>
<td>Hardware Financial</td>
</tr>
<tr>
<td>BLG0G311</td>
<td>Hardware Financial</td>
</tr>
<tr>
<td>BLG0G312</td>
<td>Hardware Financial (Secondary)</td>
</tr>
<tr>
<td>BLG0G313</td>
<td>Hardware Financial (Secondary)</td>
</tr>
<tr>
<td>BLG0N305</td>
<td>Hardware Financial - Purchase Display</td>
</tr>
<tr>
<td>BLG0D305</td>
<td>Hardware Financial - Purchase Entry</td>
</tr>
<tr>
<td>BLG0N306</td>
<td>Hardware Financial - Rental/Lease Display</td>
</tr>
<tr>
<td>BLG0D306</td>
<td>Hardware Financial - Rental/Lease Entry</td>
</tr>
<tr>
<td>BLG0N307</td>
<td>Hardware Financial - VPA Display</td>
</tr>
<tr>
<td>BLG0D307</td>
<td>Hardware Financial - VPA Entry</td>
</tr>
<tr>
<td>BLG0DU15</td>
<td>Hardware Financial Delete Verification</td>
</tr>
<tr>
<td>BLG0DU25</td>
<td>Hardware Financial Delete Verification</td>
</tr>
<tr>
<td>BLG0G391</td>
<td>Hardware Financial Description Data Inquiry</td>
</tr>
<tr>
<td>BLG0N301</td>
<td>Hardware Financial Display</td>
</tr>
<tr>
<td>BLG0N320</td>
<td>Hardware Financial Display</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>BLG0G171</td>
<td>Hardware Subcomponent Description/Control Data Inquiry</td>
</tr>
<tr>
<td>BLG0N142</td>
<td>Hardware Subcomponent Display</td>
</tr>
<tr>
<td>BLG0D140</td>
<td>Hardware Subcomponent Entry</td>
</tr>
<tr>
<td>BLG0N164</td>
<td>Hardware Subcomponent Financial Display</td>
</tr>
<tr>
<td>BLG0D164</td>
<td>Hardware Subcomponent Financial Entry</td>
</tr>
<tr>
<td>BLG0N141</td>
<td>Hardware Subcomponent Record Utilities</td>
</tr>
<tr>
<td>BLG0DU60</td>
<td>Hardware Subcomponent Rename Verification</td>
</tr>
<tr>
<td>BLG0DU33</td>
<td>Hardware Subcomponent Summary</td>
</tr>
<tr>
<td>BLG0G170</td>
<td>Hardware Subcomponent Summary</td>
</tr>
<tr>
<td>BLG0N140</td>
<td>Hardware Subcomponent Summary Display</td>
</tr>
<tr>
<td>BLG0N160</td>
<td>Hardware Subcomponent Support Display</td>
</tr>
<tr>
<td>BLG0D160</td>
<td>Hardware Subcomponent Support Entry</td>
</tr>
<tr>
<td>BLG0G172</td>
<td>Hardware Subcomponent Support/EC Levels/Financial Inquiry</td>
</tr>
<tr>
<td>BLG0DU75</td>
<td>Hardware Subcomponent Swap Verification</td>
</tr>
<tr>
<td>BLG0G135</td>
<td>Hardware+ Inquiry</td>
</tr>
<tr>
<td>BTN0ENHD</td>
<td>Help Desk Primary Options</td>
</tr>
<tr>
<td>BLG00000</td>
<td>INFO/Management Entry</td>
</tr>
<tr>
<td>BTN00000</td>
<td>INFO/Management Entry</td>
</tr>
<tr>
<td>BLG00001</td>
<td>INFO/Management Inquiry</td>
</tr>
<tr>
<td>BTN00001</td>
<td>INFO/Management Inquiry</td>
</tr>
<tr>
<td>BTN0C101</td>
<td>Initial Change Data</td>
</tr>
<tr>
<td>BTN0B102</td>
<td>Initial Problem Data</td>
</tr>
<tr>
<td>BLG0C486</td>
<td>Installation Mods</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Reference**

A-29
<table>
<thead>
<tr>
<th>Model Hardware Component</th>
<th>Model Hardware Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0DU47 Delete Verification</td>
<td>BLG0N700 Summary Display</td>
</tr>
<tr>
<td>BLG0DU48 Delete Verification</td>
<td>BLG0G710 Components</td>
</tr>
<tr>
<td>BLG0N701 Display</td>
<td>BLG0G713 Components</td>
</tr>
<tr>
<td>BLG0D700 Entry</td>
<td>BLG0G791 Description Data Inquiry</td>
</tr>
<tr>
<td>BLG0D703 Entry</td>
<td>BLG0G726 Feature</td>
</tr>
<tr>
<td>BLG0N753 Financial Display</td>
<td>BLG0DU51 Feature Delete Verification</td>
</tr>
<tr>
<td>BLG0D752 Financial Entry</td>
<td>BLG0DU54 Feature Delete Verification</td>
</tr>
<tr>
<td>BLG0D753 Financial Entry</td>
<td>BLG0N722 Feature Display</td>
</tr>
<tr>
<td>BLG0N770 Line Display</td>
<td>BLG0D704 Feature Entry</td>
</tr>
<tr>
<td>BLG0N772 Loop Display</td>
<td>BLG0D720 Feature Entry</td>
</tr>
<tr>
<td>BLG0N750 Primary Support Display</td>
<td>BLG0N754 Feature Financial Display</td>
</tr>
<tr>
<td>BLG0D750 Primary Support Entry</td>
<td>BLG0G722 Feature Display</td>
</tr>
<tr>
<td>BLG0DU66 Rename Verification</td>
<td>BLG0G727 Feature Inquiry</td>
</tr>
<tr>
<td>BLG0DU67 Rename Verification</td>
<td>BLG1TCMF Feature Record List</td>
</tr>
<tr>
<td>BLG0N710 Record Utilities</td>
<td>BLG0N721 Feature Record Utilities</td>
</tr>
<tr>
<td>BLG0D751 Secondary Support Display</td>
<td>BLG0DU38 Feature Summary</td>
</tr>
<tr>
<td>BLG0D751 Secondary Support Entry</td>
<td>BLG0G728 Feature Summary</td>
</tr>
<tr>
<td>BLG0DU36 Summary</td>
<td>BLG0N720 Feature Summary Display</td>
</tr>
<tr>
<td>BLG0DU37 Summary</td>
<td>BLG0N757 Feature Support Display</td>
</tr>
<tr>
<td>BLG0G790 Summary</td>
<td>BLG0D757 Feature Support Entry</td>
</tr>
<tr>
<td>BLG0G736 Records+</td>
<td></td>
</tr>
</tbody>
</table>

Version 1.1
<table>
<thead>
<tr>
<th>BLG0G738</th>
<th>Model Hardware Records+ Inquiry</th>
<th>BLG0G735</th>
<th>Model Hardware+ Inquiry</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0G742</td>
<td>Model Hardware Records+ Summary</td>
<td>BLG0G735</td>
<td>Model Hardware+ Inquiry</td>
</tr>
<tr>
<td>BLG0DU49</td>
<td>Model Hardware Subcomponent Delete Verification</td>
<td>BLG0DU52</td>
<td>Model Software Component Delete Verification</td>
</tr>
<tr>
<td>BLG0DU50</td>
<td>Model Hardware Subcomponent Delete Verification</td>
<td>BLG0DU53</td>
<td>Model Software Component Delete Verification</td>
</tr>
<tr>
<td>BLG0N740</td>
<td>Model Hardware Subcomponent Display</td>
<td>BLG0G771</td>
<td>Model Software Component Description/Support Data Inquiry</td>
</tr>
<tr>
<td>BLG0N742</td>
<td>Model Hardware Subcomponent Display</td>
<td>BLG0N782</td>
<td>Model Software Component Display</td>
</tr>
<tr>
<td>BLG0D740</td>
<td>Model Hardware Subcomponent Entry</td>
<td>BLG0D707</td>
<td>Model Software Component Entry</td>
</tr>
<tr>
<td>BLG0D740</td>
<td>Model Hardware Subcomponent Entry</td>
<td>BLG0D780</td>
<td>Model Software Component Entry</td>
</tr>
<tr>
<td>BLG0D764</td>
<td>Model Hardware Subcomponent Financial Entry</td>
<td>BLG0N783</td>
<td>Model Software Component Financial Display</td>
</tr>
<tr>
<td>BLG0G737</td>
<td>Model Hardware Subcomponent Inquiry</td>
<td>BLG0D783</td>
<td>Model Software Component Financial Entry</td>
</tr>
<tr>
<td>BLG1TSUB</td>
<td>Model Hardware Subcomponent List</td>
<td>BLG1TSMD</td>
<td>Model Software Component List</td>
</tr>
<tr>
<td>BLG0D760</td>
<td>Model Hardware Subcomponent Primary Support Entry</td>
<td>BLG0G772</td>
<td>Model Software Component Other Data Inquiry</td>
</tr>
<tr>
<td>BLG0N741</td>
<td>Model Hardware Subcomponent Record Utilities</td>
<td>BLG0N758</td>
<td>Model Software Component Primary Support Display</td>
</tr>
<tr>
<td>BLG0DU68</td>
<td>Model Hardware Subcomponent Rename Verification</td>
<td>BLG0D758</td>
<td>Model Software Component Primary Support Entry</td>
</tr>
<tr>
<td>BLG0DU39</td>
<td>Model Hardware Subcomponent Summary</td>
<td>BLG0N781</td>
<td>Model Software Component Record Utilities</td>
</tr>
<tr>
<td>BLG0DU49</td>
<td>Model Hardware Support/Financial/Control Data Inquiry</td>
<td>BLG0DU69</td>
<td>Model Software Component Rename Verification</td>
</tr>
<tr>
<td>BLG0G739</td>
<td>Model Hardware Subcomponent Summary</td>
<td>BLG0N759</td>
<td>Model Software Component Secondary Support Display</td>
</tr>
<tr>
<td>BLG0N760</td>
<td>Model Hardware Subcomponent Support Display</td>
<td>BLG0D759</td>
<td>Model Software Component Secondary Support Entry</td>
</tr>
<tr>
<td>BLG0G792</td>
<td>Model Hardware Support/Financial/Control Data Inquiry</td>
<td>BLG0DU44</td>
<td>Model Software Component Summary</td>
</tr>
<tr>
<td>Panel Title</td>
<td>Model Software Component/Feature</td>
<td>Panel Title</td>
<td>Model Software Component/Feature</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------</td>
<td>-------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>BLG0DU45</td>
<td>Model Software Component Summary</td>
<td>BLG0G782</td>
<td>Model Software Feature Support/Financial Data Inquiry</td>
</tr>
<tr>
<td>BLG0G770</td>
<td>Model Software Component Summary</td>
<td>BLG0G786</td>
<td>Model Software Records+</td>
</tr>
<tr>
<td>BLG0N780</td>
<td>Model Software Component Summary Display</td>
<td>BLG0G787</td>
<td>Model Software Records+ Inquiry</td>
</tr>
<tr>
<td>BLG0G750</td>
<td>Model Software Components</td>
<td>BLG0G788</td>
<td>Model Software Records+ Summary</td>
</tr>
<tr>
<td>BLG0G776</td>
<td>Model Software Feature</td>
<td>BLG0G785</td>
<td>Model Software+ Inquiry</td>
</tr>
<tr>
<td>BLG0G777</td>
<td>Model Software Feature</td>
<td>BLG0D705</td>
<td>Model Subcomponent Entry</td>
</tr>
<tr>
<td>BLG0DU55</td>
<td>Model Software Feature Delete Verification</td>
<td>BLG0G730</td>
<td>Model Subcomponent Inquiry</td>
</tr>
<tr>
<td>BLG0DU56</td>
<td>Model Software Feature Delete Verification</td>
<td>BTN0ENNO</td>
<td>Network Support Primary Options</td>
</tr>
<tr>
<td>BLG0G781</td>
<td>Model Software Feature</td>
<td>BTN0ENOM</td>
<td>Operations Manager Primary Options</td>
</tr>
<tr>
<td></td>
<td>Description Data Inquiry</td>
<td>BLG0C496</td>
<td>Optional Feature</td>
</tr>
<tr>
<td>BLG0N792</td>
<td>Model Software Feature Display</td>
<td>BLG0F496</td>
<td>Optional Feature</td>
</tr>
<tr>
<td>BLG0D709</td>
<td>Model Software Feature Entry</td>
<td>BLG0VQ74</td>
<td>Panel Flow Display</td>
</tr>
<tr>
<td>BLG0D790</td>
<td>Model Software Feature Entry</td>
<td>BLG0VE74</td>
<td>Panel Flow Inquiry</td>
</tr>
<tr>
<td>BLG0N784</td>
<td>Model Software Feature</td>
<td>BLG0V730</td>
<td>Panel Flow Update</td>
</tr>
<tr>
<td></td>
<td>Financial</td>
<td>BLM0Y310</td>
<td>Panel Set Display</td>
</tr>
<tr>
<td>BLG0D784</td>
<td>Model Software Feature Entry</td>
<td>BLM0Y101</td>
<td>Panel Set Entry</td>
</tr>
<tr>
<td>BLG0G775</td>
<td>Model Software Feature Inquiry</td>
<td>BLM0Y250</td>
<td>Panel Set Inquiry</td>
</tr>
<tr>
<td>BLG0N791</td>
<td>Model Software Feature Record Utilities</td>
<td>BLM0Y251</td>
<td>Panel Set Inquiry Summary</td>
</tr>
<tr>
<td>BLG0DU70</td>
<td>Model Software Feature Rename Verification</td>
<td>BLM0Y751</td>
<td>Parent Inquiry Summary</td>
</tr>
<tr>
<td>BLG0DU46</td>
<td>Model Software Feature Summary</td>
<td>BLM0Y700</td>
<td>Parent Record Description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0Y701</td>
<td>Parent Record Description</td>
</tr>
<tr>
<td>BLG0G780</td>
<td>Model Software Feature Summary</td>
<td>BLM0Y810</td>
<td>Parent Record Display</td>
</tr>
<tr>
<td>BLG0N790</td>
<td>Model Software Feature Summary Display</td>
<td>BLM0Y600</td>
<td>Parent Record Entry</td>
</tr>
<tr>
<td>BLG0N787</td>
<td>Model Software Feature Support Display</td>
<td>BLM0Y702</td>
<td>Parent Record Inquiry</td>
</tr>
<tr>
<td>BLG0D787</td>
<td>Model Software Feature Support Entry</td>
<td>BLM0Y703</td>
<td>Parent Record Inquiry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0YU50</td>
<td>Parent Record Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0YU51</td>
<td>Parent Record Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLM0Y740</td>
<td>Parent/Child Inquiry</td>
</tr>
</tbody>
</table>

A-32

Version 1.1
<table>
<thead>
<tr>
<th>Reference</th>
<th>Panel/Title</th>
<th>Reference</th>
<th>Panel/Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLM0Y780</td>
<td>Parent/Child Inquiry</td>
<td>BLG0E300</td>
<td>Problem Close</td>
</tr>
<tr>
<td>BLM0Y781</td>
<td>Parent/Child Inquiry Summary</td>
<td>BLG0E301</td>
<td>Problem Close</td>
</tr>
<tr>
<td>BLG0N134</td>
<td>Path Display Specification</td>
<td>BLG0E390</td>
<td>Problem Close Data Inquiry</td>
</tr>
<tr>
<td>BTN0N134</td>
<td>Path Display Specification</td>
<td>BTN0E390</td>
<td>Problem Close Data Inquiry</td>
</tr>
<tr>
<td>BLG0W200</td>
<td>Periodic Change Status Report</td>
<td>BLG0L300</td>
<td>Problem Close Display</td>
</tr>
<tr>
<td>BLG0W210</td>
<td>Periodic Problem Status Report</td>
<td>BTN0L300</td>
<td>Problem Close Display</td>
</tr>
<tr>
<td>BTN0DU18</td>
<td>Person Delete Verification</td>
<td>BLG0B300</td>
<td>Problem Close Entry</td>
</tr>
<tr>
<td>BTN0DU28</td>
<td>Person Delete Verification</td>
<td>BTN0B300</td>
<td>Problem Close Entry</td>
</tr>
<tr>
<td>BTN0DU08</td>
<td>Person Record</td>
<td>BTN0ENPC</td>
<td>Problem Controller Primary Options</td>
</tr>
<tr>
<td>BTN0D500</td>
<td>Person Record</td>
<td>BTN0EPPC</td>
<td>Problem Controller Primary Options</td>
</tr>
<tr>
<td>BTN0N500</td>
<td>Person Record</td>
<td>BTN0BU10</td>
<td>Problem Delete Verification</td>
</tr>
<tr>
<td>BTN0N510</td>
<td>Person Record</td>
<td>BTN0BU11</td>
<td>Problem Delete Verification</td>
</tr>
<tr>
<td>BTN0H000</td>
<td>Person Search Option</td>
<td>BTN0BU10</td>
<td>Problem Delete Verification</td>
</tr>
<tr>
<td>BLG0J270</td>
<td>PMF Authority</td>
<td>BTN0BU11</td>
<td>Problem Delete Verification</td>
</tr>
<tr>
<td>BLG0K270</td>
<td>PMF Authority</td>
<td>BTN0B001</td>
<td>Problem Entry</td>
</tr>
<tr>
<td>BLG0Q270</td>
<td>PMF Authority</td>
<td>BTN0B001</td>
<td>Problem Entry</td>
</tr>
<tr>
<td>BLG0EN20</td>
<td>Primary Options Menu</td>
<td>BLG0E000</td>
<td>Problem Inquiry</td>
</tr>
<tr>
<td>BLM0YE00</td>
<td>Primary Options Menu</td>
<td>BLG0E090</td>
<td>Problem Inquiry Summary</td>
</tr>
<tr>
<td>BLM0YE50</td>
<td>Primary Options Menu</td>
<td>BTN0E090</td>
<td>Problem Inquiry Summary</td>
</tr>
<tr>
<td>BLG0P420</td>
<td>Print Data Set Destination Entry</td>
<td>BLG0E700</td>
<td>Problem Record Control</td>
</tr>
<tr>
<td>BLG0P430</td>
<td>Print DDNAME Destination Entry</td>
<td>BLG0E100</td>
<td>Problem Reporter</td>
</tr>
<tr>
<td>BLG0P400</td>
<td>Print Output Destination</td>
<td>BLG0E101</td>
<td>Problem Reporter</td>
</tr>
<tr>
<td>BLG0P401</td>
<td>Print Output Destination</td>
<td>BLG0E102</td>
<td>Problem Reporter</td>
</tr>
<tr>
<td>BLG0P410</td>
<td>Print SYSPUT Destination Entry</td>
<td>BLG0E190</td>
<td>Problem Reporter Data Inquiry</td>
</tr>
<tr>
<td>BLG0J001</td>
<td>Privilege Class Entry</td>
<td>BTN0E190</td>
<td>Problem Reporter Data Inquiry</td>
</tr>
<tr>
<td>BTN0EPPC</td>
<td>Privilege Class Selection</td>
<td>BTN0L100</td>
<td>Problem Reporter Display</td>
</tr>
<tr>
<td>BLG0010</td>
<td>Privilege Class Text Entry</td>
<td>BLG0B100</td>
<td>Problem Reporter Entry</td>
</tr>
<tr>
<td>BLG0J200</td>
<td>Problem Authority</td>
<td>BLG0W020</td>
<td>Problem Reports</td>
</tr>
<tr>
<td>BLG0K200</td>
<td>Problem Authority</td>
<td>BLG0B502</td>
<td>Problem Resolution Data</td>
</tr>
<tr>
<td>BLG0Q200</td>
<td>Problem Authority</td>
<td>BLG0E590</td>
<td>Problem Resolution Data Inquiry</td>
</tr>
<tr>
<td>BTN0B101</td>
<td>Problem Caller Information</td>
<td>BLG0E200</td>
<td>Problem Status</td>
</tr>
<tr>
<td>Reference</td>
<td>Information/Management Panels Arranged by Panel Title</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLM0Y790</td>
<td>Record+ Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLM0Y791</td>
<td>Record+ Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLM0Y792</td>
<td>Record+ Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLM0Y793</td>
<td>Record+ Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0G195</td>
<td>Records+ Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG00020</td>
<td>Reference Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLGLSQMP</td>
<td>Relational Data Mapping Table Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0P300</td>
<td>Report Control Defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0W000</td>
<td>Report Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C497</td>
<td>Request Engineer Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0F497</td>
<td>Request Engineer Action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C498</td>
<td>Request Price Quotation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0F498</td>
<td>Request Price Quotation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0V710</td>
<td>Response Processing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0VQ72</td>
<td>Response Processing Display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0VE72</td>
<td>Response Processing Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG00100</td>
<td>Response Type</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0Z001</td>
<td>Rules Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0I010</td>
<td>Rules Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0I020</td>
<td>Rules Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0I030</td>
<td>Rules Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0I040</td>
<td>Rules Record Control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0Z020</td>
<td>Rules Text Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0P200</td>
<td>Screen Control Defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTN0EA90</td>
<td>Search List Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTN0EA91</td>
<td>Search List Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BTN0G19A</td>
<td>Search List Options</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU17</td>
<td>Service Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU27</td>
<td>Service Delete Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D400</td>
<td>Service Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0D401</td>
<td>Service Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0G400</td>
<td>Service Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0G411</td>
<td>Service Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0G490</td>
<td>Service Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0N410</td>
<td>Service Record Utilities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU73</td>
<td>Service Rename Verification</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0DU07</td>
<td>Service Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0G493</td>
<td>Service Summary</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0N400</td>
<td>Service Summary Display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0P100</td>
<td>Session Defaults</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLGLQHPL</td>
<td>Sharing Logical Partition Identifier Display</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLGLJHPL</td>
<td>Sharing Logical Partition Identifier Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C051</td>
<td>Software Add</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C151</td>
<td>Software Add Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C052</td>
<td>Software Alter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C053</td>
<td>Software Alter by Installing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C153</td>
<td>Software Alter by Installing Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C054</td>
<td>Software Alter by Modifying</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C154</td>
<td>Software Alter by Modifying Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C055</td>
<td>Software Alter by Removing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C155</td>
<td>Software Alter by Removing Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C152</td>
<td>Software Alter Inquiry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C412</td>
<td>Software Alteration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C482</td>
<td>Software APAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0F482</td>
<td>Software APAR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLGLPZFF</td>
<td>Software APAR Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C050</td>
<td>Software Change</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C410</td>
<td>Software Change</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reference Code</td>
<td>Panel Title</td>
<td>Reference Code</td>
<td>Panel Title</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>BLG0N232</td>
<td>Software Connection Display</td>
<td>BLG0D090</td>
<td>Software Feature Text Entry</td>
</tr>
<tr>
<td>BLG0D206</td>
<td>Software Connection Entry</td>
<td>BLGLFLDF</td>
<td>Software Field Name Entry</td>
</tr>
<tr>
<td>BLG0N231</td>
<td>Software Connection Record Utilities</td>
<td>BLG0G350</td>
<td>Software Financial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0G351</td>
<td>Software Financial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0G352</td>
<td>Software Financial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0G353</td>
<td>Software Financial</td>
</tr>
<tr>
<td>BLG0DU63</td>
<td>Software Connection Rename Verification</td>
<td>BLG0G360</td>
<td>Software Financial</td>
</tr>
<tr>
<td>BLG0DU40</td>
<td>Software Connection Summary</td>
<td>BLG0G361</td>
<td>Software Financial</td>
</tr>
<tr>
<td>BLG0N230</td>
<td>Software Connection Summary Display</td>
<td>BLG0G363</td>
<td>Software Financial</td>
</tr>
<tr>
<td>BLG0D095</td>
<td>Software Connection Text Entry</td>
<td>BLG0N353</td>
<td>Software Financial - Basic License Display</td>
</tr>
<tr>
<td>BLG0G150</td>
<td>Software Feature</td>
<td>BLG0D353</td>
<td>Software Financial - Basic License Entry</td>
</tr>
<tr>
<td>BLG0G151</td>
<td>Software Feature</td>
<td>BLG0N355</td>
<td>Software Financial - DSLO License Display</td>
</tr>
<tr>
<td>BLG0DU14</td>
<td>Software Feature Delete Verification</td>
<td>BLG0D355</td>
<td>Software Financial - DSLO License Entry</td>
</tr>
<tr>
<td>BLG0DU24</td>
<td>Software Feature Delete Verification</td>
<td>BLG0N357</td>
<td>Software Financial - VLA Display</td>
</tr>
<tr>
<td>BLG0G1B1</td>
<td>Software Feature Description/Control Data Inquiry</td>
<td>BLG0D357</td>
<td>Software Financial - VLA Entry</td>
</tr>
<tr>
<td>BLG0N222</td>
<td>Software Feature Display</td>
<td>BLG0DU16</td>
<td>Software Financial Delete Verification</td>
</tr>
<tr>
<td>BLG0D203</td>
<td>Software Feature Entry</td>
<td>BLG0DU26</td>
<td>Software Financial Delete Verification</td>
</tr>
<tr>
<td>BLG0D220</td>
<td>Software Feature Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0N254</td>
<td>Software Feature Financial Display</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0G357</td>
<td>Software Financial Description Data Inquiry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0N352</td>
<td>Software Financial Display</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0D350</td>
<td>Software Financial Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0D356</td>
<td>Software Financial Entry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0G355</td>
<td>Software Financial Inquiry</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0N360</td>
<td>Software Financial Record Utilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0DU04</td>
<td>Software Feature Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0DU65</td>
<td>Software Financial Rename Verification</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0DU20</td>
<td>Software Feature Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0DU06</td>
<td>Software Financial Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0N257</td>
<td>Software Feature Support Display</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0G358</td>
<td>Software Financial Summary</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BLG0D257</td>
<td></td>
</tr>
</tbody>
</table>

Information/Management Panels Arranged by Panel Title
<table>
<thead>
<tr>
<th>Panel</th>
<th>Description</th>
<th>Panel</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0N350</td>
<td>Software Financial Summary Display</td>
<td>BLG0V000</td>
<td>SRC Inquiry</td>
</tr>
<tr>
<td>BLG0D055</td>
<td>Software Financial Text Entry</td>
<td>BLG0V800</td>
<td>SRC Record Control</td>
</tr>
<tr>
<td>BLG0G362</td>
<td>Software Financial VLA</td>
<td>BLG0V801</td>
<td>SRC Start Panel</td>
</tr>
<tr>
<td>BLG0G359</td>
<td>Software Financial VLA/Control Data Inquiry</td>
<td>BLG0U000</td>
<td>SRC Summary</td>
</tr>
<tr>
<td>BLG0C481</td>
<td>Software Level</td>
<td>BLG0U001</td>
<td>SRC Summary</td>
</tr>
<tr>
<td>BLG0F481</td>
<td>Software Level</td>
<td>BLG0V090</td>
<td>SRC Summary</td>
</tr>
<tr>
<td>BLG0G166</td>
<td>Software Records+</td>
<td>BLG0T100</td>
<td>SRC Summary Display</td>
</tr>
<tr>
<td>BLG0G167</td>
<td>Software Records+ Inquiry</td>
<td>BLG0P520</td>
<td>Standard Report Data Set Destination Entry</td>
</tr>
<tr>
<td>BLG0G168</td>
<td>Software Records+ Summary</td>
<td>BLG0P521</td>
<td>Standard Report Data Set Destination Entry</td>
</tr>
<tr>
<td>BLG0C056</td>
<td>Software Remove</td>
<td>BLG0P530</td>
<td>Standard Report DDNAME Destination Entry</td>
</tr>
<tr>
<td>BLG0C156</td>
<td>Software Remove Inquiry</td>
<td>BLG0P531</td>
<td>Standard Report DDNAME Destination Entry</td>
</tr>
<tr>
<td>BLG0G1A2</td>
<td>Software Support/FMID/Financial Inquiry</td>
<td>BLG0P500</td>
<td>Standard Report Destination</td>
</tr>
<tr>
<td>BLG0G165</td>
<td>Software+ Inquiry</td>
<td>BLG0P501</td>
<td>Standard Report Output Destination</td>
</tr>
<tr>
<td>BLG0G1C0</td>
<td>Source Information</td>
<td>BLG0P502</td>
<td>Standard Report Output Destination</td>
</tr>
<tr>
<td>BLG0D107</td>
<td>Specification for Hardware Model Copy</td>
<td>BLG0P510</td>
<td>Standard Report SYSOUT Destination Entry</td>
</tr>
<tr>
<td>BLG0D204</td>
<td>Specification for Software Model Copy</td>
<td>BLG0P511</td>
<td>Standard Report SYSOUT Destination Entry</td>
</tr>
<tr>
<td>BLG0V802</td>
<td>SRC Application</td>
<td>BLG0VQ73</td>
<td>Structured Word and Validation Display</td>
</tr>
<tr>
<td>BLG0J290</td>
<td>SRC Authority</td>
<td>BLG0V720</td>
<td>Structured Word and Validation Collection</td>
</tr>
<tr>
<td>BLG0K290</td>
<td>SRC Authority</td>
<td>BLG0VE73</td>
<td>Structured Word and Validation Inquiry</td>
</tr>
<tr>
<td>BLG0Q290</td>
<td>SRC Authority</td>
<td>BLG0D141</td>
<td>Subcomponent Entry</td>
</tr>
<tr>
<td>BLG0TU01</td>
<td>SRC Delete Verification</td>
<td>BLG0G130</td>
<td>Subcomponent Inquiry</td>
</tr>
<tr>
<td>BLG0TU02</td>
<td>SRC Delete Verification</td>
<td>BTN0ENSU</td>
<td>Support Primary Options</td>
</tr>
<tr>
<td>BLG0T110</td>
<td>SRC Description Display</td>
<td>BTN0ENSY</td>
<td>System Administrator Primary Options</td>
</tr>
<tr>
<td>BLG0U100</td>
<td>SRC Description Entry</td>
<td>BTN0ENSY</td>
<td>System Administrator Primary Options</td>
</tr>
<tr>
<td>BLG0V100</td>
<td>SRC Description Inquiry</td>
<td>BTN0ENSY</td>
<td>System Administrator Primary Options</td>
</tr>
<tr>
<td>BLG0V190</td>
<td>SRC Description Inquiry</td>
<td>BTN0ENSY</td>
<td>System Administrator Primary Options</td>
</tr>
<tr>
<td>BLG0U101</td>
<td>SRC Entry</td>
<td>BTN0ENSY</td>
<td>System Administrator Primary Options</td>
</tr>
<tr>
<td>BLG0T200</td>
<td>SRC Execution Classes</td>
<td>BTN0ENSY</td>
<td>System Administrator Primary Options</td>
</tr>
<tr>
<td>BLG0U200</td>
<td>SRC Execution Classes</td>
<td>BTN0ENSY</td>
<td>System Administrator Primary Options</td>
</tr>
</tbody>
</table>

Version 1.1
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLG0K292</td>
<td>System Class Authority</td>
<td>BLG0F485</td>
<td>Upgrade Subsets</td>
</tr>
<tr>
<td>BLG0DU19</td>
<td>System Delete Verification</td>
<td>BLG0C484</td>
<td>Upgrade/Tape</td>
</tr>
<tr>
<td>BLG0DU29</td>
<td>System Delete Verification</td>
<td>BLG0F484</td>
<td>Upgrade/Tape</td>
</tr>
<tr>
<td>BLG0D600</td>
<td>System Entry</td>
<td>BLG0B901</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG0D601</td>
<td>System Entry</td>
<td>BLG0B902</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG0G290</td>
<td>System Inquiry</td>
<td>BLG0B903</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG0G850</td>
<td>System Inquiry</td>
<td>BLG0L901</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG00011</td>
<td>System Record Inquiry</td>
<td>BLG0L902</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG0N610</td>
<td>System Record Utilities</td>
<td>BLG0L903</td>
<td>Users to Notify</td>
</tr>
<tr>
<td>BLG0G851</td>
<td>System Records</td>
<td>BLG0VQ52</td>
<td>Validation Control Data Display</td>
</tr>
<tr>
<td>BLG0DU72</td>
<td>System Rename Verification</td>
<td>BLG0VLST</td>
<td>Validation Data Display</td>
</tr>
<tr>
<td>BLG0DU09</td>
<td>System Summary</td>
<td>BLG0V502</td>
<td>Validation Display Update</td>
</tr>
<tr>
<td>BLG0G293</td>
<td>System Summary</td>
<td>BLG0VD50</td>
<td>Validation Record Delete Verification</td>
</tr>
<tr>
<td>BLG0N600</td>
<td>System Summary Display</td>
<td>BLG0VD51</td>
<td>Validation Record Delete Verification</td>
</tr>
<tr>
<td>BLG0D010</td>
<td>Text Entry</td>
<td>BLG0V500</td>
<td>Validation Record Entry</td>
</tr>
<tr>
<td>BLG0D020</td>
<td>Text Entry</td>
<td>BLG0V501</td>
<td>Validation Record Entry</td>
</tr>
<tr>
<td>BLG0D025</td>
<td>Text Entry</td>
<td>BLG0V510</td>
<td>Validation Record Inquiry</td>
</tr>
<tr>
<td>BLG0D030</td>
<td>Text Entry</td>
<td>BLG0VU50</td>
<td>Validation Summary</td>
</tr>
<tr>
<td>BLG0D040</td>
<td>Text Entry</td>
<td>BLG0VQ50</td>
<td>Validation Summary Display</td>
</tr>
<tr>
<td>BLG0D060</td>
<td>Text Entry</td>
<td>BLG00102</td>
<td>Verification</td>
</tr>
<tr>
<td>BLG0D520</td>
<td>Text Entry</td>
<td>BLG0BU90</td>
<td>Verification of Users Record Delete</td>
</tr>
<tr>
<td>BLG0D620</td>
<td>Text Entry</td>
<td>BLG00101</td>
<td>Verify Change Operation</td>
</tr>
<tr>
<td>BLM0Y120</td>
<td>Text Entry</td>
<td>BLG00104</td>
<td>Verify Change Operation</td>
</tr>
<tr>
<td>BLM0Y620</td>
<td>Text Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLM0Y630</td>
<td>Text Entry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLG0C485</td>
<td>Upgrade Subsets</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Training Records

Use the panels that follow as a guide when creating training records. The panels show data to enter in each record. To access these panels, enter the immediate response chain (IRC) 5,1 on the command line of Primary Options Menu panel for the Management application. Once you create the first record, you can copy and modify it to create the rest.

Create training records so that users can follow the training exercises in the User's Guide. To create training records, you must be authorized to create problem records.

Creating Training Records

TRN850

<table>
<thead>
<tr>
<th>BLG08100</th>
<th>PROBLEM REPORTER ENTRY</th>
<th>PROBLEM: TRN850</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Enter problem reporter data; cursor placement or input line entry allowed.</td>
<td></td>
</tr>
<tr>
<td>1. Reported by......&lt;R&gt; THOREAU_______</td>
<td>13. Problem type.........</td>
<td></td>
</tr>
<tr>
<td>2. Reporter dept....... 76 ______</td>
<td>14. Problem status....&lt;R&gt; INITIAL</td>
<td></td>
</tr>
<tr>
<td>3. Reporter phone...... 555-6767____</td>
<td>15. User problem number.. TRN850</td>
<td></td>
</tr>
<tr>
<td>4. Date occurred........ ______</td>
<td>16. Initial priority.....</td>
<td></td>
</tr>
<tr>
<td>5. Time occurred........ ____</td>
<td>17. Outage................</td>
<td></td>
</tr>
<tr>
<td>6. Network name......... ______</td>
<td>18. Rerun time...........</td>
<td></td>
</tr>
<tr>
<td>7. System name......... ______</td>
<td>19. Network impact........</td>
<td></td>
</tr>
<tr>
<td>8. Program name....... XEDIT____</td>
<td>20. System impact........</td>
<td></td>
</tr>
<tr>
<td>10. Key item affected.... ______</td>
<td>22. Device impact........</td>
<td></td>
</tr>
<tr>
<td>11. Date fix required... 10/19/97</td>
<td>23. User form number.....</td>
<td></td>
</tr>
<tr>
<td>12. Time fix required... ____</td>
<td>24. Location code........ ROOM890</td>
<td></td>
</tr>
<tr>
<td>25. Description......&lt;R&gt; RESPONSE TIME TOO SLOW (TRAINING RECORD)____</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When you finish, type END to save or CANCEL to discard any changes.
TRN376

<table>
<thead>
<tr>
<th>BLG08100</th>
<th>PROBLEM REPORTER ENTRY</th>
<th>PROBLEM: TRN376</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enter problem reporter data; cursor placement or input line entry allowed.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Reported by......&lt;R&gt; YAGER__________</td>
<td>13. Problem type........</td>
<td></td>
</tr>
<tr>
<td>2. Reporter dept....... ___________</td>
<td>14. Problem status....&lt;R&gt; CLOSED</td>
<td></td>
</tr>
<tr>
<td>3. Reporter phone....... _____________</td>
<td>15. User problem number.. TRN376__</td>
<td></td>
</tr>
<tr>
<td>4. Date occurred....... ________</td>
<td>16. Initial priority..... __</td>
<td></td>
</tr>
<tr>
<td>5. Time occurred....... _____</td>
<td>17. Outage............... ________</td>
<td></td>
</tr>
<tr>
<td>6. Network name........ ________</td>
<td>18. Rerun time........... ________</td>
<td></td>
</tr>
<tr>
<td>7. System name......... ________</td>
<td>19. Network impact........ ________</td>
<td></td>
</tr>
<tr>
<td>8. Program name........ ________</td>
<td>20. System impact........ ________</td>
<td></td>
</tr>
<tr>
<td>10. Key item affected... ________</td>
<td>22. Device impact......... ________</td>
<td></td>
</tr>
<tr>
<td>11. Date fix required... ________</td>
<td>23. User form number..... ________</td>
<td></td>
</tr>
<tr>
<td>12. Time fix required... _____</td>
<td>24. Location code........ ROOM876_</td>
<td></td>
</tr>
<tr>
<td>25. Description......&lt;R&gt; NEEDS ANOTHER DISK DRIVE (TRAINING RECORD)___</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

```
TRN744

Enter problem reporter data; cursor placement or input line entry allowed.

1. Reported by......<R> HOLLOWAY_______ 13. Problem type...........
2. Reporter dept....... ___________ 14. Problem status....<R> OPEN_
3. Reporter phone...... ___________ 15. User problem number.. TRN744_
4. Date occurred........_______ 16. Initial priority......
5. Time occurred........_______ 17. Outage..............
6. Network name....... ________ 18. Rerun time.........
7. System name........ ________ 19. Network impact......
8. Program name....... ISPF____ 20. System impact......
10. Key item affected... ________ 22. Device impact........
11. Date fix required... ________ 23. User form number....
12. Time fix required... _____ 24. Location code........ ROOM444_
13. Problem type...........
14. Problem status....<R> OPEN_
15. User problem number.. TRN744_
16. Initial priority......
17. Outage..............
18. Rerun time.........
19. Network impact......
20. System impact......
21. Program impact......
22. Device impact........
23. User form number....
24. Location code........ ROOM444_

25. Description......<R> CANNOT ACCESS APPLICATIONS (TRAINING RECORD)_

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

TRN100

Enter problem reporter data; cursor placement or input line entry allowed.

1. Reported by......<R> THOREAU________ 13. Problem type...........
2. Reporter dept....... ___________ 14. Problem status....<R> INITIAL_
3. Reporter phone...... ___________ 15. User problem number.. TRN100_
4. Date occurred........_______ 16. Initial priority......
5. Time occurred........_______ 17. Outage..............
6. Network name....... ________ 18. Rerun time.........
7. System name........ ________ 19. Network impact......
8. Program name....... ________ 20. System impact......
10. Key item affected... ________ 22. Device impact........
11. Date fix required... ________ 23. User form number....
12. Time fix required... _____ 24. Location code........ ROOM89_/.
13. Problem type...........
14. Problem status....<R> INITIAL_
15. User problem number.. TRN100_
16. Initial priority......
17. Outage..............
18. Rerun time.........
19. Network impact......
20. System impact......
21. Program impact......
22. Device impact........
23. User form number....
24. Location code........ ROOM89_/.

25. Description......<R> BROKEN ON/OFF SWITCH (TRAINING RECORD)_______

When you finish, type END to save or CANCEL to discard any changes.
Creating Training Records

TRN200

BLG08100  PROBLEM REPORTER ENTRY  PROBLEM: TRN200

Enter problem reporter data; cursor placement or input line entry allowed.

1. Reported by......<R> HOLLOWAY_______
2. Reporter dept....... ___________
3. Reporter phone....... _____________
4. Date occurred....... _____________
5. Time occurred....... ________
6. Network name........ ________
7. System name......... ________
8. Program name........ ________
9. Device name......... 3278____
10. Key item affected... ________
11. Date fix required... ________
12. Time fix required... _____
13. Problem type........
14. Problem status....<R> INITIAL
15. User problem number.. TRN200___
16. Initial priority..... __
17. Outage............... ________
18. Rerun time........... ________
19. Network impact....... ________
20. System impact....... ________
21. Program impact...... ________
22. Device impact........ ________
23. User form number..... ________
24. Location code........ ROOM444_
25. Description......<R> COLOR CONVERGENCE PROBLEM (TRAINING RECORD)___

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

===>

Version 1.1
TRN300

BLG08100  PROBLEM REPORTER ENTRY  PROBLEM: TRN300

Enter problem reporter data; cursor placement or input line entry allowed.

1. Reported by......<R> HARRISON________ 13. Problem type........
2. Reporter dept....... ___________ 14. Problem status.......<R> CLOSED
3. Reporter phone....... ___________ 15. User problem number.. TRN300
4. Date occurred....... ________ 16. Initial priority..... __
5. Time occurred....... ________ 17. Outage................. __
6. Network name....... ________ 18. Rerun time............. ________
7. System name....... ________ 19. Network impact......... ________
8. Program name....... ________ 20. System impact........ ________
9. Device name....... 6670 21. Program impact........ ________
10. Key item affected.... ________ 22. Device impact........ ________
11. Date fix required... ________ 23. User form number...... ________
12. Time fix required... ________ 24. Location code........ ROOM679
13. Date fix required... ________ 25. Description......<R> NEEDS TONER MAKES STREAKS (TRAINING RECORD)________

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

====>

TRN400

BLG08100  PROBLEM REPORTER ENTRY  PROBLEM: TRN400

Enter problem reporter data; cursor placement or input line entry allowed.

1. Reported by......<R> HARRISON________ 13. Problem type........
2. Reporter dept....... ___________ 14. Problem status.......<R> OPEN
3. Reporter phone....... ___________ 15. User problem number.. TRN400
4. Date occurred....... ________ 16. Initial priority..... __
5. Time occurred....... _____ 17. Outage............... ________
6. Network name....... ________ 18. Rerun time............. ________
7. System name....... ________ 19. Network impact......... ________
8. Program name....... ________ 20. System impact........ ________
9. Device name....... 1403 21. Program impact........ ________
10. Key item affected.... ________ 22. Device impact........ ________
11. Date fix required... ________ 23. User form number...... ________
12. Time fix required... ________ 24. Location code........ COMPROOM
13. Date fix required... ________ 25. Description......<R> PRINTING TOO LIGHT (TRAINING RECORD)________

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

====>
TRN500

<table>
<thead>
<tr>
<th>BLG08100</th>
<th>PROBLEM REPORTER ENTRY</th>
<th>PROBLEM: TRN500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter problem reporter data; cursor placement or input line entry allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Reported by......&lt;R&gt; YAGER__________</td>
<td>13. Problem type........</td>
<td></td>
</tr>
<tr>
<td>2. Reporter dept....... ___________</td>
<td>14. Problem status.....&lt;R&gt; OPEN__</td>
<td></td>
</tr>
<tr>
<td>3. Reporter phone....... ___________</td>
<td>15. User problem number.. TRN500__</td>
<td></td>
</tr>
<tr>
<td>4. Date occurred....... ________</td>
<td>16. Initial priority..... __</td>
<td></td>
</tr>
<tr>
<td>5. Time occurred....... _____</td>
<td>17. Outage.................</td>
<td></td>
</tr>
<tr>
<td>6. Network name......... ________</td>
<td>18. Rerun time.............</td>
<td></td>
</tr>
<tr>
<td>7. System name......... ________</td>
<td>19. Network impact........</td>
<td></td>
</tr>
<tr>
<td>8. Program name........ XEDIT__</td>
<td>20. System impact........</td>
<td></td>
</tr>
<tr>
<td>10. Key item affected... ________</td>
<td>22. Device impact.........</td>
<td></td>
</tr>
<tr>
<td>11. Date fix required... ________</td>
<td>23. User form number.....</td>
<td></td>
</tr>
<tr>
<td>12. Time fix required... _____</td>
<td>24. Location code........ ROOM876_</td>
<td></td>
</tr>
<tr>
<td>25. Description......&lt;R&gt; CANNOT BLOCKCOPY TEXT (TRAINING RECORD)_____</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

```
### TRN700

<table>
<thead>
<tr>
<th>BLG08100</th>
<th>PROBLEM REPORTER ENTRY</th>
<th>PROBLEM: TRN700</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter problem reporter data; cursor placement or input line entry allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Reported by......&lt;R&gt; KOWALSKI</td>
<td>13. Problem type........</td>
<td></td>
</tr>
<tr>
<td>2. Reporter dept........</td>
<td>14. Problem status ......&lt;R&gt; INITIAL</td>
<td></td>
</tr>
<tr>
<td>3. Reporter phone...... 555-6789</td>
<td>15. User problem number .. TRN700</td>
<td></td>
</tr>
<tr>
<td>4. Date occurred........</td>
<td>16. Initial priority ......</td>
<td></td>
</tr>
<tr>
<td>5. Time occurred........</td>
<td>17. Outage.................</td>
<td></td>
</tr>
<tr>
<td>6. Network name........</td>
<td>18. Rerun time.............</td>
<td></td>
</tr>
<tr>
<td>10. Key item affected...</td>
<td>22. Device impact...........</td>
<td></td>
</tr>
<tr>
<td>11. Date fix required...</td>
<td>23. User form number........</td>
<td></td>
</tr>
<tr>
<td>12. Time fix required...</td>
<td>24. Location code........... ROOM678</td>
<td></td>
</tr>
<tr>
<td>13. Problem type........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. Problem status ......&lt;R&gt; INITIAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. User problem number .. TRN700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. Initial priority ......</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17. Outage.................</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18. Rerun time.............</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19. Network impact..........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20. System impact..........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21. Program impact..........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22. Device impact..........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23. User form number........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24. Location code........... ROOM678</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

25. Description......<R> PAPER JAM CANNOT BE FIXED (TRAINING RECORD) ____________

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

### TRN800

<table>
<thead>
<tr>
<th>BLG08100</th>
<th>PROBLEM REPORTER ENTRY</th>
<th>PROBLEM: TRN800</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter problem reporter data; cursor placement or input line entry allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Reported by......&lt;R&gt; KEENE</td>
<td>13. Problem type........</td>
<td></td>
</tr>
<tr>
<td>2. Reporter dept........</td>
<td>14. Problem status ......&lt;R&gt; INITIAL</td>
<td></td>
</tr>
<tr>
<td>3. Reporter phone...... 555-1234</td>
<td>15. User problem number .. TRN800</td>
<td></td>
</tr>
<tr>
<td>4. Date occurred........</td>
<td>16. Initial priority ......</td>
<td></td>
</tr>
<tr>
<td>5. Time occurred........</td>
<td>17. Outage.................</td>
<td></td>
</tr>
<tr>
<td>6. Network name........</td>
<td>18. Rerun time.............</td>
<td></td>
</tr>
<tr>
<td>10. Key item affected...</td>
<td>22. Device impact...........</td>
<td></td>
</tr>
<tr>
<td>11. Date fix required...</td>
<td>23. User form number........</td>
<td></td>
</tr>
<tr>
<td>12. Time fix required...</td>
<td>24. Location code........... ROOM333</td>
<td></td>
</tr>
<tr>
<td>13. Problem type........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14. Problem status ......&lt;R&gt; INITIAL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. User problem number .. TRN800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. Initial priority ......</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17. Outage.................</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18. Rerun time.............</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19. Network impact..........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20. System impact..........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21. Program impact..........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22. Device impact..........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23. User form number........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24. Location code........... ROOM333</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

25. Description......<R> BROKEN DISK DRIVE (TRAINING RECORD) ____________

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

```
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

When you finish, type END to save or CANCEL to discard any changes.
### TRN559

<table>
<thead>
<tr>
<th>BLG08100</th>
<th>PROBLEM REPORTER ENTRY</th>
<th>PROBLEM: TRN559</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter problem reporter data; cursor placement or input line entry allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Reported by......&lt;R&gt; BONANO_________    13. Problem type...........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Reporter dept....... _________________    14. Problem status...&lt;R&gt; CLOSED_</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Reporter phone....... _________________    15. User problem number.. TRN559__</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Date occurred....... _________________    16. Initial priority.....</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Time occurred....... _________________    17. Outage...............</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Network name......... _________________    18. Rerun time...........</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 7. System name......... _________________ &nbsp;&nbsp;&nbsp;19. Network impact....
| 8. Program name......... _________________ &nbsp;&nbsp;&nbsp;20. System impact..... |
| 9. Device name......... 1403_________ &nbsp;&nbsp;&nbsp;21. Program impact...... |
| 10. Key item affected... _________________ &nbsp;&nbsp;&nbsp;22. Device impact........ |
| 11. Date fix required... _________________ &nbsp;&nbsp;&nbsp;23. User form number.... |
| 12. Time fix required... _________________ &nbsp;&nbsp;&nbsp;24. Location code....... COMPROOM |
| 25. Description......<R> GARBAGE ON MY PRINTOUTS (TRAINING RECORD)____ |

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

>>>}

### TRN125

<table>
<thead>
<tr>
<th>BLG08100</th>
<th>PROBLEM REPORTER ENTRY</th>
<th>PROBLEM: TRN125</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter problem reporter data; cursor placement or input line entry allowed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Reported by......&lt;R&gt; KOWALSKI_______    13. Problem type...........</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2. Reporter dept....... _________________    14. Problem status...&lt;R&gt; OPEN___</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Reporter phone....... _________________    15. User problem number.. TRN125__</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Date occurred....... _________________    16. Initial priority.....</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. Time occurred....... _________________    17. Outage...............</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6. Network name......... _________________    18. Rerun time...........</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 7. System name......... _________________ &nbsp;&nbsp;&nbsp;19. Network impact....
| 8. Program name........ APPLXYZ_ &nbsp;&nbsp;&nbsp;20. System impact..... |
| 9. Device name......... _________________ &nbsp;&nbsp;&nbsp;21. Program impact...... |
| 10. Key item affected... _________________ &nbsp;&nbsp;&nbsp;22. Device impact........ |
| 11. Date fix required... _________________ &nbsp;&nbsp;&nbsp;23. User form number.... |
| 12. Time fix required... _________________ &nbsp;&nbsp;&nbsp;24. Location code....... ROOM678_ |
| 25. Description......<R> DOES NOT FILL UP SCREEN (TRAINING RECORD)____ |

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

>>>
<table>
<thead>
<tr>
<th>Field</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reported by</td>
<td>BONANO</td>
</tr>
<tr>
<td>Reporter dept</td>
<td></td>
</tr>
<tr>
<td>Reporter phone</td>
<td></td>
</tr>
<tr>
<td>Date occurred</td>
<td></td>
</tr>
<tr>
<td>Time occurred</td>
<td></td>
</tr>
<tr>
<td>Network name</td>
<td></td>
</tr>
<tr>
<td>System name</td>
<td>ISPF</td>
</tr>
<tr>
<td>Key item affected</td>
<td></td>
</tr>
<tr>
<td>Date fix required</td>
<td></td>
</tr>
<tr>
<td>Time fix required</td>
<td></td>
</tr>
<tr>
<td>Problem type</td>
<td></td>
</tr>
<tr>
<td>Problem status</td>
<td>OPEN</td>
</tr>
<tr>
<td>User problem number</td>
<td>TRN600</td>
</tr>
<tr>
<td>Initial priority</td>
<td></td>
</tr>
<tr>
<td>Outage</td>
<td></td>
</tr>
<tr>
<td>Rerun time</td>
<td></td>
</tr>
<tr>
<td>System impact</td>
<td></td>
</tr>
<tr>
<td>Program impact</td>
<td></td>
</tr>
<tr>
<td>Device impact</td>
<td></td>
</tr>
<tr>
<td>User form number</td>
<td></td>
</tr>
<tr>
<td>Location code</td>
<td>ROOM890</td>
</tr>
<tr>
<td>Outage type</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>PASSWORD DOES NOT WORK (TRAINING RECORD)</td>
</tr>
</tbody>
</table>

When you finish, type END to save or CANCEL to discard any changes.
## TRN366

<table>
<thead>
<tr>
<th>BLG08100</th>
<th>PROBLEM REPORTER ENTRY</th>
<th>PROBLEM: TRN366</th>
</tr>
</thead>
</table>

Enter problem reporter data; cursor placement or input line entry allowed.

| 1. Reported by......<R> GARTLAND ______ | 13. Problem type....... |
| 2. Reporter dept....... ___________ | 14. Problem status......<R> INITIAL |
| 3. Reporter phone...... 555-1234_____ | 15. User problem number.. TRN366 |
| 4. Date occurred....... ________ | 16. Initial priority...... |
| 5. Time occurred....... ________ | 17. Outage................. |
| 6. Network name....... ________ | 18. Rerun time............ |
| 7. System name....... ________ | 19. Network impact........ |
| 8. Program name....... APPLXYZ_ | 20. System impact........ |
| 9. Device name....... ________ | 21. Program impact........ |
| 10. Key item affected... ________ | 22. Device impact........ |
| 11. Date fix required... ________ | 23. User form number..... |
| 12. Time fix required... ________ | 24. Location code........ ROOM333_ |
| 13. Problem type....... | 25. Description......<R> WILL NOT INITIALIZE (TRAINING RECORD) ______ |

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

===>
Your Organization’s Procedures

This chapter can help you to keep track of the local procedures and conventions that your organization follows. You can record this information on the charts provided and refer to them in the future.

Before you start, record the name and telephone number of your program administrator for quick reference.

| Local Contact: |
| Program Administrator: |
| ________________________________ |
| Telephone |
| Number:__________________________ |

Getting Started

Use the procedure below to get started on Information/Management. Record the CLIST set up by your organization on the line provided.

Starting Information/Management:
1. Log on to MVS.

2. Choose ISPF option number:_______________
   or
   Enter CLIST named: ____________________________________________
Numbering a Record

Each record must have a unique record ID. Your organization may already have a procedure for assigning ID numbers to records. If so, write that procedure below. To find out what that procedure is, call your program administrator.

Record ID Numbering Procedure:

Note: If you do not assign an ID to a record, Information/Management automatically assigns a numeric ID.

Data-Entry Conventions

Just as your organization may have standard numbering procedures for records, it may also have standard procedures for entering specific kinds of data in the database.

Having a standard method for entering similar kinds of data makes it easier to search for data after it is stored in the database. For example, if you entered someone’s name as JOHNSMITH, then next time entered JSMITH, and again entered SMITHJ, it would be difficult to find all records that contained the name JOHN SMITH.

Your organization recognizes the importance of having data-entry conventions. Some of the panels on which you enter data have fields that accept data in a specific format and, therefore, do not let you enter data improperly. However, this type of data checking cannot be done on all types of data. Therefore, your organization has established data-entry conventions for you to follow when entering data into records. Write the conventions on the next page so you can easily refer to them whenever you enter data.
### Data-entry conventions:

**Names:**

______________________________________________________________

______________________________________________________________

**Dates:**

______________________________________________________________

______________________________________________________________

**Times:**

______________________________________________________________

______________________________________________________________

**Product names:**

______________________________________________________________

______________________________________________________________

**Machine types:**

______________________________________________________________

______________________________________________________________

**Location codes:**

______________________________________________________________

______________________________________________________________

**Department names:**

______________________________________________________________

______________________________________________________________

**Other data:**

______________________________________________________________

______________________________________________________________

______________________________________________________________

______________________________________________________________

______________________________________________________________

______________________________________________________________

______________________________________________________________
Deleting a Record

Your organization may have a local procedure that you should follow to delete a record from the database. If so, write that procedure here.

Local procedure for getting a record deleted:

_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
_________________________________________________________________
# Printing a Record

Your organization may have local printing defaults that are used in printing a record. If so, write those defaults here.

<table>
<thead>
<tr>
<th>Local printing defaults:</th>
</tr>
</thead>
<tbody>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
<tr>
<td>________________________</td>
</tr>
</tbody>
</table>
Relating Publications to Specific Tasks

Your data processing organization can have many different users performing many different tasks. The books in the Information/Management 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 Information/Management publication that supports those tasks. See “The Information/Management Library” on page E-1 for more information about each book.

## Typical Tasks

<table>
<thead>
<tr>
<th>If You Are:</th>
<th>And You Do This:</th>
<th>Read This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluating Information/Management</td>
<td>Learn about Information/Management. Evaluate and assess its benefits to your organization.</td>
<td>General Information</td>
</tr>
<tr>
<td>Planning to Use Information/Management</td>
<td>Identify the hardware and software requirements of Information/Management. Identify the prerequisite and corequisite products. Plan and implement a test system.</td>
<td>Planning and Installation Guide and Reference</td>
</tr>
<tr>
<td>Installing Information/Management</td>
<td>Install Information/Management. Define and initialize data sets. Create session-parameters members.</td>
<td>Planning and Installation Guide and Reference</td>
</tr>
<tr>
<td></td>
<td>Define and create multiple Information/Management BLX-SPs.</td>
<td>Planning and Installation Guide and Reference</td>
</tr>
<tr>
<td></td>
<td>Define and create APPC transaction programs for clients. Define and create APPC transaction programs for shared DASD.</td>
<td>Client Installation and User's Guide</td>
</tr>
<tr>
<td>Diagnosing problems</td>
<td>Diagnose problems encountered while using Information/Management</td>
<td>Diagnosis Guide</td>
</tr>
<tr>
<td>If You Are:</td>
<td>And You Do This:</td>
<td>Read This:</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Administering Information/Management</td>
<td>Manage user profiles and passwords. Define and maintain privilege class records. Define and maintain rules records.</td>
<td>Program Administration Guide and Reference Integration Facility Guide</td>
</tr>
<tr>
<td></td>
<td>Define and maintain USERS record. Define and maintain ALIAS record.</td>
<td>Program Administration Guide and Reference</td>
</tr>
<tr>
<td></td>
<td>Implement GUI interface. Define and maintain command aliases and authorizations.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Implement and administer Notification Management. Create user-defined line commands. Define logical database partitioning.</td>
<td>Program Administration Guide and Reference</td>
</tr>
<tr>
<td>Maintaining Information/Management</td>
<td>Set up access to the data sets. Maintain the databases. Define and maintain privilege class records.</td>
<td>Planning and Installation Guide and Reference Program Administration Guide and Reference</td>
</tr>
<tr>
<td></td>
<td>Define and maintain the BLX-SP. Run the utility programs.</td>
<td>Operation and Maintenance Reference</td>
</tr>
<tr>
<td>Programming applications</td>
<td>Use the application program interfaces.</td>
<td>Application Program Interface Guide</td>
</tr>
<tr>
<td></td>
<td>Use the application program interfaces for Information/Management clients.</td>
<td>Client Installation and User’s Guide</td>
</tr>
<tr>
<td></td>
<td>Create web applications using or accessing Information/Management data.</td>
<td>World Wide Web Interface Guide</td>
</tr>
</tbody>
</table>
### Table 41 (Page 3 of 3). Relating Publications to Specific Tasks

<table>
<thead>
<tr>
<th>If You Are:</th>
<th>And You Do This:</th>
<th>Read This:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customizing Information/Management</td>
<td>Design and implement a Change Management system. Design and implement a Configuration Management system. Design and implement a Problem Management system.</td>
<td>Problem, Change, and Configuration Management</td>
</tr>
<tr>
<td></td>
<td>Design, create, and test terminal simulator panels or terminal simulator EXECs. Customize panels and panel flow.</td>
<td>Terminal Simulator Guide and Reference</td>
</tr>
<tr>
<td></td>
<td>Design, create, and test Information/Management formatted reports.</td>
<td>Report Format Facility Guide and Reference</td>
</tr>
<tr>
<td></td>
<td>Create a bridge between NetView and Information/Management applications. Integrate Information/Management with Tivoli distributed products.</td>
<td>NetView and TME 10 Interfaces Reference</td>
</tr>
<tr>
<td>Assisting Users</td>
<td>Create, search, update, and close change, configuration, or problem records. Browse or print Change, Configuration, or Problem Management reports.</td>
<td>Problem, Change, and Configuration Management</td>
</tr>
<tr>
<td></td>
<td>Use the Information/Management Integration Facility.</td>
<td>Integration Facility Guide</td>
</tr>
<tr>
<td>Using Information/Management</td>
<td>Obtain an overall view of Information/Management.</td>
<td>General Information</td>
</tr>
<tr>
<td></td>
<td>Learn about the Information/Management panel types, record types, and commands. Change a user profile.</td>
<td>User’s Guide</td>
</tr>
<tr>
<td></td>
<td>Learn about Problem, Change, and Configuration Management records.</td>
<td>Problem, Change, and Configuration Management</td>
</tr>
<tr>
<td></td>
<td>Receive and respond to Information/Management messages.</td>
<td>Messages and Codes</td>
</tr>
<tr>
<td></td>
<td>Design and create reports.</td>
<td>Report Format Facility Guide and Reference</td>
</tr>
</tbody>
</table>
Where to Find More Information

The Information/Management library is an integral part of Information/Management. The books are written with particular audiences in mind. Each book covers specific tasks. See “Relating Publications to Specific Tasks” on page D-1 to find out which book refers to which task.

A subscription to the Information/Management customer newsletter, *The Structured Word*, is available to you free of charge. Send your complete mailing address to:

Tivoli Systems
The Structured Word
Dept. X7GA, Bldg. 503
P.O. Box 12195
3039 Cornwallis Road
Research Triangle Park, NC 27709-2195
USA

In the United States, Information/Management courses are available through the IBM Education Center to help you learn to use the product. See “Information/Management Courses” on page F-1 for more information about them. Outside of the United States, contact your local IBM representative for information concerning courses available to you.

The Information/Management Library

The publications shipped automatically with each Information/Management Version 1.1 licensed program are:

- **Application Program Interface Guide**
- **Client Installation and User’s Guide**
- **Diagnosis Guide**
- **Licensed Program Specification**
- **Integration Facility Guide**
- **Master Index, Glossary, and Bibliography**
- **Messages and Codes**
- **NetView and TME 10 Interfaces Reference**
- **Operation and Maintenance Reference**
- **Panel Modification Facility Guide**
- **Planning and Installation Guide and Reference**
- **Program Administration Guide and Reference**
Problem, Change, and Configuration Management
Reference Summary
Report Format Facility Guide and Reference
Terminal Simulator Guide and Reference
User’s Guide
World Wide Web Interface Guide.

Also included is the Product Kit, which includes the complete online library on CD-ROM.

Additional copies of these items are available for a fee. The remaining books in the library are also available for a fee:

General Information

Request these items from your IBM representative or the IBM 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 Information/Management library.

To order a set of publications, specify order number SBOF-8559-00.

Application Program Interface Guide, SC34-4592-00, 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.

Client Installation and User’s Guide, SC34-4593-00, describes and illustrates the setup and use of Information/Management’s remote clients. This book shows you how to use Information/Management functions in the AIX, CICS, HP-UX, OS/2, Sun Solaris, and Windows NT environments. Also included in this book is complete information about using the Information/Management servers.

Diagnosis Guide, SC34-4594-00, explains how to identify a problem, analyze its symptoms, and resolve it. This book includes tools and information that are helpful in solving problems found when you use Information/Management.

The General Information brochure, GC34-4595-00, gives an overview of the uses, capabilities, and advantages of Information/Management. It also gives an overview of what is new for this version of Information/Management.

Integration Facility Guide, SC34-4596-00, explains the concepts and structure of the Integration Facility. The Integration Facility provides a task-oriented interface to Information/Management that makes the Information/Management applications easier to use. This book also explains how to use the panels and panel flows in your change and problem management system.

Master Index, Glossary, and Bibliography, SC34-4598-00, combines the indexes from each book in the Information/Management library for Version 1.1. Also included is a complete glossary and bibliography for the product.
Messages and Codes, GC34-4599-00, contains the messages and completion codes issued by the various Information/Management applications. Each entry includes an explanation of the message or code and recommends actions for users and system programmers.

NetView and TME 10 Interfaces Reference, SC34-4613-00, describes the steps to follow to make an automatic connection between NetView and Information/Management applications. It also explains how to customize the application interface which serves as an application enabler for the NetView Bridge. (The NetView AutoBridge Guide and Reference, formerly a separate document, is now included in this manual.) Information on interfaces to the Tivoli Management Environment is also provided.

Operation and Maintenance Reference, SC34-4601-00, 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 Information/Management licensed program, Version 1.1.

Panel Modification Facility Guide, SC34-4602-00, gives detailed instructions for creating and modifying Information/Management panels. It provides detailed checklists for the common panel modification tasks, and it provides reference information useful to those who design and modify panels.

Planning and Installation Guide and Reference, SC34-4603-00, describes the tasks required for installing Information/Management. This book contains:

- A chapter with a step-by-step overview that guides you through planning and installing Information/Management.
- A section describing the many Information/Management functions and features that you must consider for your installation.
- A section helping you to plan the resources that Information/Management requires.
- A section describing the migration tasks that must be performed if you are currently using another version of Information/Management.
- A section explaining the tasks that you must perform to install, tailor, and start Information/Management Version 1.1.
- Appendixes with reference material that help you in installing Information/Management.

Problem, Change, and Configuration Management, SC34-4604-00, 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.

Program Administration Guide and Reference, SC34-4605-00, provides detailed information about Information/Management program administration tasks, such as defining user profiles and privilege classes, and enabling the GUI user interface.

Reference Summary, SC34-4607-00, is a reference booklet containing Information/Management commands, a list of p-words and s-words, summary
information for PMF, and other information you need when you use Information/Management.

*Report Format Facility Guide and Reference*, SC34-4608-00, explains how to modify standard Information/Management reports and create new ones. It describes and illustrates the syntax of Report Format Facility statements, which are used in report format tables (RFTs) to define the output from the REPORT and PRINT commands. It also includes several examples of modified RFTs.

*Terminal Simulator Guide and Reference*, SC34-4609-00, explains how to use terminal simulator panels (TSPs) and EXECs (TSXs) that let you simulate an entire interactive session with an Information/Management 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.

*User’s Guide*, SC34-4610-00, provides a general introduction to Information/Management and databases. This book has a series of step-by-step exercises, showing beginning users how to copy, update, print, create, and delete records, and how to search a database. It also contains Information/Management command syntax and descriptions, and other reference information.

*World Wide Web Interface Guide*, SC34-4570-00, explains how to install and operate the features available with Information/Management that enable you to access an Information/Management database using a Web browser as a client.
Information/Management Courses

Education Offerings

Information/Management 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 Information/Management. 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). Or, on the World Wide Web, visit http://www.training.ibm.com.

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:

c kontakt educ uk@vnet.ibm.com

On the World Wide Web, visit:

http://www.europe.ibm.com/education-uk

- **Information/Management for End Users (EA76):** You will learn how to enter data, search the database (several ways), use prefix words, work with problem, change, and configuration records, display privilege class records, update your profile, run standard reports, and write and use stored response chains.

- **Customising Information/Management (EA77):** You will learn how to customise Information/Management by using the Panel Modification facility to move, delete, and create new fields. You will also learn how to create new search results lists, and modify and create any panel.
Education Offerings

- **Advanced Customising Information/Management (EA78):** A follow-on to EA77, in four days you will further learn to customise Information/Management. You will create a new record type and diagnose and correct PMF and dialog problems as well as learn more about panel control information and more advanced coding of control and terminal simulator panels.

- **Customising Information/Management Reports (EA79):** You will learn to write reports (without requiring PMF skill), print out any data in any record, use nested searches, create variables, calculate totals, perform date and time arithmetic, understand s-words and p-words, print history information and free-form text, use DO loops, call supplied exits, and produce colour graphic reports on the terminal.
The following related documents might be of interest and help to you as you use Information/Management.

**Advanced Interactive Executive (AIX)**
- *AIX SNA Server/6000: Command Reference*, SC31-7100
- *AIX SNA Server/6000: Configuration Reference*, SC31-7014
- *AIX SNA Server/6000: Diagnosis Guide and Messages*, SC31-7101
- *AIX SNA Server/6000: Transaction Program Reference*, SC31-7003
- *AIX Version 3.2 Files Reference*, GC23-2200
- *AIX Version 3.2 System Management Guide: Communications and Networks*, GC23-2487
- *AIX Version 3.2: Commands Reference*, GC23-2366

**BookManager**
- *BookManager READ/MVS General Information*, GC38-2032
- *BookManager READ/MVS Getting Started*, SC38-2033
- *BookManager READ/MVS Displaying Online Books*, SC38-2034
- *BookManager READ/MVS Installation, Planning, and Customization*, SC38-2035

**Assembler H**
- *Assembler H Version 2 Language Reference*, GC26-4037

**Common Programming Interface Communications (CPIC)**
- *Common Programming Interface Communications CPI-C Reference*, SC26-4399

**Communications Management (CM/2)**
- *Communications Manager/2 Configuration Guide*, SC31-6171
Customer Information Control System (CICS)
CICS/ESA Intercommunication Guide, SC33-0657
CICS/ESA Installation Guide, SC33-0663
CICS/ESA System Definition Guide, SC33-0664
CICS/ESA Customization Guide, SC33-0665
CICS/ESA Resource Definition (Online), SC33-0666
CICS/ESA Resource Definition (Macro), SC33-0667
CICS/ESA CICS-Supplied Transactions, SC33-0669
CICS/ESA System Programming Reference, SC33-0670

DATABASE 2 (DB2)
IBM Database 2 Version 3 Administration Guide, SC26-4888
IBM Database 2 Version 3 SQL Reference, SC26-4890
IBM Database 2 Version 3 Messages and Codes, SC26-4892
IBM Database 2 Version 3 Command and Utility Reference, SC26-4891

Graphical Data Display Manager (GDDM)
GDDM Application Programming Guide, SC33-0337
GDDM Base Programming Reference, SC33-0332
GDDM Diagnosis and Problem Determination Guide, SC33-0326
GDDM General Information, GC33-0319
GDDM Guide for Users, SC33-0327
GDDM Image Symbol Editor, SC33-0329
GDDM Installation and System Management for MVS, GC33-0321
GDDM-PGF Interactive Chart Utility, SC33-0328
GDDM Interactive Map Definition, SC33-0338
GDDM Messages, SC33-0325
GDDM Performance Guide, SC33-0324
GDDM-PGF Programming Reference, SC33-0333
GDDM Vector Symbol Editor, SC33-0330

Interactive System Productivity Facility (ISPF) Version 4.1
ISPF User’s Guide, SC34-4484-00
ISPF Dialog Developer’s Guide and Reference, SC34-4486-00
ISPF Dialog Tag Language Guide and Reference, SC34-4441-00
ISPF Edit and Edit Macros, SC34-4446-00
ISPF Planning and Customizing, SC34-4443-00
Interactive System Productivity Facility (ISPF) Version 4.2
ISPF User’s Guide, SC34-4484-01
ISPF Dialog Developer’s Guide and Reference, SC34-4486-01
ISPF Dialog Tag Language Guide and Reference, SC34-4441-01
ISPF Edit and Edit Macros, SC34-4446-01
ISPF Planning and Customizing, SC34-4443-01

Interactive System Productivity Facility (ISPF) Version 4.2.1
ISPF User’s Guide, SC28-1239
ISPF Dialog Developer’s Guide and Reference, SC28-1273
ISPF Dialog Tag Language Guide and Reference, SC28-1219
ISPF Edit and Edit Macros, SC28-1312
ISPF Planning and Customizing, SC28-1298

Multiple Virtual Storage (MVS)
See also the documents listed for OS/390.

APPCC System Definitions in MVS/ESA and OS/2, GG66-3224
DFDSS User’s Guide, SC26-4388
DFSMSS/MVS Access Method Services for the Integrated Catalog Facility, SC26-4906
DFSMSS/MVS Macro Instructions for Data Sets, SC26-4913
DFSMSS/MVS Storage Administration Guide for DFSMSdss Storage, SC26-4930
DFSMSS/MVS Using Data Sets, SC26-4922
MVS/DFP Access Method Services for the Integrated Catalog Facility, SC26-4562
MVS/DFP Access Method Services for VSAM Catalogs, SC26-4570
MVS/DFP Diagnosis Reference, LY27-9571
MVS/DFP Macro Instructions for VSAM Data Sets, SC26-4569
MVS/DFP Using Data Sets, SC26-4749
MVS/ESA APPC/MVS Handbook for the OS/2 System Administrator, GC28-1133
MVS/ESA Application Development: Writing Transaction Programs for APPC/MVS, GC28-1121
MVS/ESA Application Development: Writing Servers for APPC/MVS, GC28-1070
MVS/ESA Diagnosis: Reference, LY28-1820
MVS/ESA Diagnosis: Tools and Service Aids, LY28-1813
MVS/ESA Initialization and Tuning Reference, GC28-1635
MVS/ESA JCL Reference, GC28-1654
MVS/ESA Planning: Global Resource Serialization, GC28-1818
MVS/ESA Planning: APPC Management, GC28-1110
MVS/ESA Planning: Sysplex Management, GC28-1620
MVS/ESA SP Installation Exits, GC28-1637
MVS/ESA System Commands, GC28-1626
MVS/ESA System Management Facilities (SMF), GC28-1628
MVS/ESA VSAM Administration: Macro Instruction Reference for MVS/DFP, SC26-4517

NetView
NetView Administration Reference, SC31-7080
NetView Application Programming Guide, SC31-7081
NetView Bridge Implementation, SC31-6131
NetView Command Quick Reference, SX75-0090
NetView Customization Guide, SC31-7091
NetView Customization: Using PL/I and C, SC31-7093
NetView Customization: Writing Command Lists, SC31-7092
NetView Problem Determination and Diagnosis, LY43-0101
NetView Installation and Administration Guide, SC31-7084
NetView Messages, SC31-7096
NetView Operation, SC31-7066
Network General Information, GC31-7098
Network Automation Planning, SC31-7080
Network Storage Estimates, SK2T-6016

Operations Planning and Control/Enterprise Systems Architecture (OPC/ESA)
OPC/ESA General Information, GH19-6715
Getting Started with OPC/ESA, SH19-4060
OPC/ESA Installation Guide, SH19-4010
OPC/ESA Customization and Tuning, SH19-4011
OPC/ESA Messages and Codes, SH19-6719
OPC/ESA Planning and Scheduling the Workload, SH19-4012
OPC/ESA Controlling and Monitoring the Workload, SH19-4013
OPC/ESA Workload Monitor/2 User’s Guide, SH19-6847
OPC/ESA Programming Interfaces, SH19-4014
OPC/ESA Quick Reference, GH19-6885
OPC/ESA Diagnosis Guide and Reference, LY19-6350
OPC Tracker Agent for AIX, Installation and Operation, LY19-6396
OPC Tracker Agent for OS/400, Installation and Operation, LY19-6397

Operating System 2 (OS/2)
CM/2 V1 System Management Programming Reference, SC31-6173
CM/2 V1 Application Programming Guide, SC31-7012
Integrating OS/2 Workstations into Local Area Networks and Enterprise Networks, GG22-9490

OS/390
Publications are available for OS/390 Version 1 Release 1, 2, and 3 and most related MVS products in the IBM Online Library Omnibus Edition: OS/390 Collection, SK2T-6700. You can view the contents of the collection on the Internet at http://www.elink.ibmlink.ibm.com/pbl/pbl. You can view the booklet that accompanies this collection on IBMLINK under the category IBMMANUALS.

Programming Language One (PL/I)
OS PL/I Programming: Language Reference, SC26-4308

Problem, Change, System, and Configuration Management
Systems Management Process Model: Managing Problems and Changes, GG24-1733

Resource Access Control Facility (RACF)
RACF Command Language Reference, SC28-0733
RACF Macros and Interfaces, SC28-1345
Resource Access Control Facility Support for MVS/ESA SP 4.2, GC28-1075
RACF V1 R9.2 Security Administrator’s Guide, SC28-1340
System Programming Library: Resource Access Control Facility, SC28-1343

Resource Measurement Facility (RMF)
RMF User’s Guide, GC33-6483
RMF Diagnosis Guide, LY33-9177

Service Level Reporter (SLR)
SLR Command and Macro Reference, SH19-6532
SLR Data Base Table Reference, SH19-6533
SLR Diagnosis Guide and References, LY19-6253
SLR Installation and Customization, SH19-6437
SLR Messages and ABEND Codes, SH19-6534
SLR Tailoring Guide, SH19-6531
SLR User’s Guide: Dialogs, SH19-6530
SAA SystemView
An Introduction to SystemView, GC23-0576
Concepts, SC23-0578
SAA Common Programming Interface Communications Reference, SC26-4399
Systems Network Architecture (SNA)
SNA Network Architecture LU 6.2 Reference: Peer Protocol, SC31-6808
SNA Technical Overview, GC30-3073
Tivoli Management Environment (TME 10)
TME 10 Enterprise Console User's Guide, GC31-8506
TME 10 Global Enterprise Manager Application Policy Manager User's Guide, GC31-5108
TME 10 Global Enterprise Manager Installation and User's Guide, GC31-8474
TME 10 Global Enterprise Manager Instrumentation Guide, GC31-5109
TME 10 Global Enterprise Manager Problem Service User's Guide, GC31-8543
TME 10 NetView for OS/390 Installation and Administration Guide, SC31-8236
TME 10 Software Distribution User's Guide, GC31-8330
Transmission Control Protocol/Internet Protocol (TCP/IP)
IBM TCP/IP Version 2 for OS/2: Installation and Administration, SC31-6075
IBM TCP/IP for MVS: Application Programming Interface Reference, SC31-7187
IBM TCP/IP for MVS: Customization and Administration Guide, SC31-7134
IBM TCP/IP for MVS: Diagnosis Guide, LY43-0105
IBM TCP/IP for MVS: Messages and Codes, SC31-7132
IBM TCP/IP for MVS: Network Print Facility, SC31-8074
IBM TCP/IP for MVS: Offloading TCP/IP Processing, SC31-7133
IBM TCP/IP for MVS: Planning and Migration Guide, SC31-7189
IBM TCP/IP for MVS: Programmer’s Reference, SC31-7135
IBM TCP/IP for MVS: Quick Reference, SX75-0095
IBM TCP/IP for MVS: User’s Guide, SC31-7136
Time Sharing Option (TSO)

*TSO Extensions V2 CLISTs, SC28-1876*
*TSO Extensions REXX Reference, SC28-1883*
*TSO Extensions REXX User's Guide, SC28-1882*
*TSO Extensions Programming Services, SC28-1875*
*TSO Extensions Terminal Messages, GC28-1310*

Virtual Telecommunications Access Method (VTAM)

*VTAM Diagnosis, LY43-0065*
*VTAM Network Implementation Guide, SC31-6494*
*VTAM Programming for LU 6.2, SC31-6437*
*VTAM Resource Definition Reference, SC31-6498*
This glossary contains definitions of data processing terms that might be unfamiliar to you. It includes terms and definitions from the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

**A**

**ABEND.** Abnormal end.

**abnormal end (ABEND).** Termination of a task before its completion because of an error condition that cannot be resolved by recovery facilities while the task is running.

**Access Method Services (AMS).** A utility program that defines VSAM data sets (or files) and allocates space for them, converts indexed sequential data sets to key-sequenced data sets with indexes, modifies data set attributes in the catalog, facilitates data set portability between operating systems, creates backup copies of data sets and indexes, helps make inaccessible data sets accessible, and lists data set records and catalog entries.

**action bar.** In Common User Access architecture, the area at the top of a panel that contains choices that give a user access to actions available in that panel.

**activity record.** Activity records describe the activities associated with a change. Change activities could include ordering, installing, testing, or updating documentation as you update software.

**administrative history entry (AHE).** A data area that represents those data items from the structured database panels that request tracking. Generally, the items collected represent responses to fields that have the journal flag on.

**advanced program-to-program communication (APPC).** An implementation of the SNA/SDLC LU6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

**AHE.** Administrative history entry.

**alphanumeric data.** For Information/Management, alphanumeric data includes the characters A-Z and 0-9. Lowercase letters are not included.

**AMS.** Access Method Services.

**anchor.** In a linked list, the first block of data that contains an anchor pointer to another block of data.

**anchor pointer.** A pointer used by a called procedure to access a block of data that was created by another routine from a package of routines.

**APAR.** Authorized program analysis report.
API. Application program interface.

APB. AHE parameter block.

APL. A Programming Language.

APPC. Advanced program-to-program communication.

Applet. An application program, written in the Java programming language, that can be retrieved from a web server and executed by a web browser. A reference to an applet appears in the markup for a Web page, in the same way that a reference to a graphics file appears; a browser retrieves an applet in the same way that it retrieves a graphics file.

Application Policy Manager (APM). In the TME 10 Global Enterprise Manager (GEM), a function that enables a TME 10 administrator to monitor, control, and configure distributed applications residing in the TME 10 workstation environment or OS/390 environment through a graphical user interface (GUI).

Application program interface (API). The formally defined programming language interface between an IBM system control program or licensed program and the user of the program.

Assisted-entry panel. A panel that tells you how to enter data by providing the format and examples.

Associated data. A definition given to an item of data in a record that is to be returned as part of an inquiry result.

Asynchronous propagation. The propagation of data at a later time, not within the same unit of work as the update call. See also data propagation.

Authority code. A code value used in response validation. It adds privilege class processing to determine whether a field should be processed.

Authorized library. A library that may contain authorized programs.

Authorized program. A system program or user program that is allowed to use restricted functions.

Authorized program analysis report (APAR). A request for correction of a problem caused by a defect in a current unaltered release of a program.

Automatic data. Data, such as date, time, user ID, and certain profile data, that is retrieved automatically by Information/Management when requested by entering an equal sign (=).

Automatic Log Save Facility. An Information/Management facility that consists of the Automatic Log Save Send and Receive functions. This facility enables propagation of Information/Management data to another Information/Management database.

Automatic Log Save Receive function. This function receives send data sets from an Information/Management server running the Automatic Log Save Send function and adds the records in the data set to the database.

Automatic Log Save Send function. This function manages the offloading of the SDLDS to the send data set and sends this data set to the Information/Management server running the Automatic Log Save Receive function.

Autotask. This is a NetView operator station task that is not associated with a terminal and involves no logged-on operator. An autotask can automatically issue commands or command lists as a result of either a message table instruction or a timer.
B

basic partitioned access method (BPAM). An access method that can be applied to create program libraries in direct-access storage for convenient storage and retrieval of programs.

basic sequential access method (BSAM). An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential-access or a direct-access device.

batch message processing (BMP). In IMS/VS, a batch processing program that has access to online databases and message queues.

BIAS. Buffer invalidation address space.

BLX Service Provider (BLX-SP). A central address space database server. A central clearinghouse for I/O processing. See also central address space.

BLX-SP. BLX Service Provider.

BMP. Batch message processing.

BookManager. A program that enables you to build, display, search, and manage online books.

BPAM. Basic partitioned access method.

bridge dispatcher component. This internal component is responsible for distributing the transaction requests to multiple database servers.

bridge requester APIs component. This internal component contains the APIs to request a transaction and to extract data from the transaction reply.

BSAM. Basic sequential access method.

buffer invalidation address space (BIAS). An MVS/ESA address space that receives, via APPC/MVS, VSAM buffer invalidation requests.

C

call. (1) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point. (2) To transfer control to a procedure, program, routine, or subroutine. (3) In data communication, the actions necessary to make a connection between two stations. (4) To attempt to contact a user, regardless of whether the attempt is successful.

CALL command exit routine (CCE). An exit routine used with the CALL command statement in the Report Format Facility.

CCB. Communications control block.

CCE. CALL command exit routine.

central address space. An area in storage that does all the I/O processing (and other processing) for the user’s address spaces. With the addition of multiserver support in Information/Management, it is possible to run more than one central address space at any one time on an MVS-based system. See also BLX Service Provider.

CF. Communication facility.
change data. Information that is identified, collected, tracked, and analyzed using the Change Management facility.

Change Management facility. A facility of Information/Management that enables you to plan and organize changes in a methodical way.

child record. A type of record that is logically associated with a parent record. A child record contains specific information that relates to the information in the parent record. A child record cannot exist without a parent record.

CICS. Customer Information Control System.

CICS conversational mode. A mode of operation of a computer system in which a sequence of alternating entries and responses between a user and the system takes place like a dialog between two persons. During this dialog, the application holds the locks and the storage resources until the next user response. Contrast with CICS pseudoconversational mode.

CICS pseudoconversational mode. The CICS term for running an application in segmented mode. A series of single CICS transactions designed to appear to the operator as a continuous conversation. Contrast with CICS conversational mode.

client. (1) User. (2) A functional unit that receives shared services from a server.

client/server. In TCP/IP, the model of interaction in distributed data processing in which a program at one site sends a request to a program at another site and awaits a response. The requesting program is called a client; the answering program is called a server.

client conversation processor. A program that connects a client conversation to a HLAPI session. Transactions from the client flow through this connection to the HLAPI. The results from the HLAPI flow back through this connection to the client.

CLIST. Command list.

CM/2. Communications Manager/2.

CMDMDL. Command model statements.

cognize. The means by which Information/Management enables an organization to control whether entered data is searchable. The Cognize response field enables an organization to control which words are entered in the SDIDS as searchable words. Cognized data is searchable; other data that is not cognized is not searchable.

combined search. A type of search that helps you limit the scope of the search and tailor the results specifically to your needs. You begin defining a search argument through a structured or quick search and complete the argument with a freeform search.

command. A request from a terminal for the performance of an operation or the running of a particular program.

command list (CLIST). A data set or a member of a partitioned data set containing TSO commands to be performed sequentially in response to the EXEC command.

command model statement (CMDMDL). Needed for each NetView command used by the Bridge product.

command procedure. Either a command processor written in a high-level language or a command list.

command processor. In the NetView program, a user-written module designed to perform a specific function. Command processors, which can be written in assembler or a high-level language, are started as commands.
command statements. In a report format table, they control record processing such as retrieving, formatting and printing. PUT and SEARCH are command statements. The Report Format Facility processes control statements during the output processing phase.

Common User Access (CUA) architecture. Guidelines for the dialog between a human and a workstation or terminal.

communication facility (CF). Transmission capability, or the means for providing such capability, made available by a communication common carrier or by a telecommunication administration.

communications control block (CCB). A storage area used by a computer program to hold communications information.

Communications Manager/2 (CM/2). An OS/2 program that lets a workstation connect to a host computer and use the host resources as well as the resources of other personal computers to which the workstation is attached, either directly or through a host system. CM/2 also provides application program interfaces (APIs) so that users can develop their own applications.

component description file (CDF). In the context of the Application Management Specification (AMS), an application description file that contains information about a specific component in a management-ready application. Each management-ready application can contain multiple components, each of which is represented by one component description file.

component record. A hardware component record defines and describes a particular device that is part of a system. A software component record describes a software program running on, or available on, your system.

configuration data. Data concerning the manner in which the hardware and software of an information processing system are organized and interconnected.

configuration diagram. A collection of subdiagrams, each of which is a picture of a network of hardware components and all connected components.

Configuration Management facility. A facility of Information/Management that enables you to maintain an up-to-date and readily accessible inventory of your software and hardware configuration.

connection record. A record that describes the connection between two components. The components connected can be hardware components, software components, or both.

connectivity path. The path between two components.

control block. A storage area used by a computer program to hold control information.

control data. (1) Information about how a transaction or panel is to be processed. (2) Information used by the HLAPI to set up the environment that is used for an invocation of the Information/Management Application Program Interface.

control line. A unit of a panel structure used to contain control information that directs the processing of a selection, response field, or panel flow.

control panel. A multipurpose panel that can perform functions such as adding response data, testing for response data, altering the panel flow, calling program exits and TSPs, and storing information to be used by Information/Management functions.

control section (CSECT). The part of a program specified by the programmer to be a relocatable unit, all elements of which are to be loaded into adjoining main storage locations.
conversation. In APPC, the communications between the application program and another application program at the remote system.

conversational mode. See CICS conversational mode.

CSECT. Control section.

CUA. Common User Access.

Customer Information Control System (CICS). An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

D

DAB. Dialog anchor block.

DASD. Direct-access storage device.

database. A collection of data with a given structure for accepting, storing, and providing data for multiple users.

database record (DBR). (1) The collection of data segments using a common key in the SDDS data set. (2) In IMS/VS, a collection of segments that contains one occurrence of the root segment type and all of its dependents arranged in a hierarchical sequence. It may be smaller than, equal in size to, or larger than the access method logical record.

data attribute record. A record which describes the data and its attributes.

data buffer. A storage area where response data is stored. See also response buffer.

data control block (DCB). A control block used by access method routines in storing and retrieving data.

data definition name (ddname). The name of a data definition statement that corresponds to a data control block that contains the same name.

data-entry panel. A panel in which the user communicates with the system by filling in one or more fields.

data model record. A record which describes the composition of a data record, that is, the fields which define a record type and the attributes of each of those fields.

data propagation. The application of changes to one set of data to the copy of that data in another database system or in another Information/Management database. See also asynchronous propagation.

data set name (dsname). The term or phrase used to identify a data set.

data view record. A record which contains data attribute RNIDs. A data view record can only be used by APIs.

DBR. Database record.

DBCS. Double-byte character set.

DB2 Extract Facility. An Information/Management facility that consists of the DB2 Extract Send Utility and the DB2 Update Utilities. It converts records in a send data set into SQL statements for loading into DB2. This facility enables propagation of Information/Management data to DB2.
**DB2 Extract Send Utility.** An Information/Management utility that consists of the Information/Management SQL Generator and the programs that call it, and the programs that start the DB2 Update Utility.

**DB2 Update Utility.** An Information/Management utility that consists of the JCL and the programs that are responsible for loading Information/Management data into DB2.

**DCB.** Data control block.

**ddname.** Data definition name.

**definition statements.** In a Report Format Table, they define segments of data that remain constant either for the whole Report Format Table or for a section. TITLE and HEADING are definition statements. The Report Format Facility processes definition statements during the input processing phase.

**DFSORT.** An IBM-licensed sort merge program.

**dialog.** In an interactive system, a series of related inquiries and responses similar to a conversation between two people.

**Dialog Tag Language (DTL).** A tag-based markup language used to define the elements that are used by ISPF as the user interface tool for ISPF applications.

**dialog anchor block (DAB).** The control block that is the anchor for the chain of SDEs that make up a structured dialog.

**dictionary data set.** A data set that contains the definitions for the p-words, s-words, and validation patterns associated with Information/Management data.

**dictionary data set record map (GDDR).** Maps the records in the dictionary data set. There are two mappings, one for structured words and one for prefix records.

**direct-access storage device (DASD).** A device in which access time is effectively independent of the location of the data.

**direct add item.** An item of data that is collected by a control panel using an add control line.

**disk operating system (DOS).** An operating system for computer systems that use disks and diskettes for auxiliary storage of programs and data.

**distributed systems license option (DSLO).** A license option available to IBM customers with a basic license that permits them to copy certain IBM-licensed materials for the purpose of installing multiple systems.

**domain.** In SNA, a system services control point (SSCP) and the physical units, logical units, links, link stations, and all the associated resources that the SSCP has the ability to control by means of activation requests and deactivation requests.

**DOS.** Disk operating system.

**double-byte character set (DBCS).** Some languages have too many symbols to be able to represent all the characters using one-byte code (SBCS). To create coded character sets for such languages, each character requires 2 bytes. A DBCS blank character consists of the 2 bytes X'4040'. A DBCS null character consists of the 2 bytes X'0000'. Other DBCS characters consist of 2 bytes, each within the range X'41'-X'FE'. In data, DBCS data is preceded by a shift out (SO) character; the data is followed by a shift in (SI) character. Contrast with SBCS. See also shift out (SO) and shift in (SI).

**draw function.** The draw function enables you to create a configuration diagram from the data set produced by the REPORT command.
DSLO. Distributed systems license option.
dsnme. Data set name.
DTL. Dialog Tag Language.
dynamic PIDT. A PIDT containing entry rows built by the LLAPI on the retrieve
transaction. A dynamic PIDT contains an entry row for each SDE in the Management
application record. A dynamic PIDT can be used in subsequent create or update
transactions.

E

EC. Engineering change.
end of file (EOF). A coded character recorded on a data medium to indicate the end of
the medium.

enhanced panel style. Displays Information/Management panels with an action bar on
the first line to enable users to select pull-down menus. Programmable function key
definitions are supplied that are specific to particular Information/Management tasks. See
also action bar, pull-down menu, and function key. Contrast with standard panel style.

enterprise. A term used to generally describe a work-related organization that includes
one or more computer systems. This organization could be a company, a group of
companies, a division or subdivision of a company, a university, or some other similar type
of organization.

enterprise-developed commands. Command lists (CLISTs), REXX programs, or
high-level language programs (PL/I or C) written to customize the NetView program by an
enterprise. These commands can be written to request transactions from a database such as
Information/Management.

EOF. End of file.

escalation control block (ESCB). The place where information is stored until the time
when a control that, when set, triggers notification to specified users who track or resolve
issues as the installation determines. See also escalation facility, notification facility.

escalation facility. A part of Information/Management’s notification facility, the
escalation facility enables you to set criteria and send alert messages to whomever you
designate and at intervals you specify, until the problem is resolved.

ESCB. Escalation control block.

exit routine. (1) A routine that receives control when a specified event occurs, such as
an error. (2) Any of several types of user-written routines.

extended addressing mode. A direct-addressing mode that can access any area in
storage.

extension. Additional equipment on the same line and on the same premises, but at a
location other than the main station.
F

field. On a data medium, a specified area used for a particular class of data.

FILEDEF. A command that identifies the input or output files to be used when a program is running.

flatten. To copy a file to a buffer from an Information/Management database or a database that has the same format as an Information/Management database. After flattening, the flattened data can be saved in a file, on tape, or transmitted to a remote system. The flattened data cannot be accessed by Information/Management; to access the data, you must restore the record using the UNFLATTEN control line.

freeform. Pertaining to entry of data or the coding of statements without regard for predefined formats.

freeform argument. A definition given to a freeform search keyword used in inquiry processing that is entered on the command line along with the search command, or a simulation of freeform keyword search processing.

freeform search. The type of search in which you issue the search command and specify a search argument with this command.

freeform text entry (FTE). A data area containing fields that represent data entered by the user in response to a freeform text panel.

freeform word. A word entered without regard for predefined formats.

function key. A key that performs a specified set of operations when it is pressed.

FTE. Freeform text entry.

G

GDDM. Graphical Data Display Manager.

GDDR. Dictionary data set record map.

Generalized Trace Facility (GTF). An optional OS/VS service program that records significant system events, such as supervisor calls and start I/O operations, for the purpose of problem determination.

global resource serialization (GRS). A facility required to serialize access to data sets shared between processors.

global resource serialization complex. (1) A group of MVS systems set up to use the facilities provided by the GRS. (2) Consists of one or more systems connected by communication links.

Graphical Data Display Manager (GDDM). (1) A group of routines that enables pictures to be defined and displayed procedurally through function routines that correspond to graphic primitives. (2) An IBM licensed program that creates page segments.

group items. Record entries that have multiple p-words associated with a particular data item.

GRS. Global resource serialization.

GTF. Generalized Trace Facility.
heartbeat. A monitor for instrumented components.

help panel. Information displayed by a system in response to a help request from a user.

high-level application program interface (high-level API or HLAPI). A program interface that enables user-written programs to access selected Information/Management functions. It is generally easier to use than the LLAPI, but it provides less control. It uses the LLAPI to process requested transactions. See also low-level application program interface.

High-Level Application Program Interface/AIX (HLAPI/AIX). A program interface installed on an AIX workstation that enables user-written programs to access selected Information/Management functions installed on a host.

High-Level Application Program Interface/HP (HLAPI/HP). A program interface installed on an HP UNIX workstation that enables user-written programs to access selected Information/Management functions installed on a host.

High-Level Application Program Interface/NT (HLAPI/NT). A program interface installed on a Windows NT workstation that enables user-written programs to access selected Information/Management functions installed on a host.

High-Level Application Program Interface/Solaris (HLAPI/Solaris). A program interface installed on a Sun Solaris workstation that enables user-written programs to access selected Information/Management functions installed on a host.

High-Level Application Program Interface/UNIX (HLAPI/UNIX). The generic name for the AIX, HP, and Solaris clients.

High-Level Application Program Interface/2 (HLAPI/2). A program interface installed on a workstation that enables user-written programs to access selected Information/Management functions installed on a host.

high-level language (HLL). A programming language that does not reflect the structure of any particular computer or operating system.

high-level language command or program. A command or program written in a high-level language, such as PL/I or C.

HLAPI. High-level application program interface (high-level API).

HLAPI/AIX. High-level application program interface (high-level API) that runs on the AIX environment.

HLAPI/HP. High-level application program interface (high-level API) that runs on the HP UNIX environment.

HLAPI/NT. A program interface installed on a Windows NT workstation that enables user-written programs to access selected Information/Management functions installed on a host.

HLAPI/REXX interface. A function that allows a REXX program to use HLAPI transactions.

HLAPI/Solaris. High-level application program interface (high-level API) that runs on the Sun Solaris environment.

HLAPI/2. High-level application program interface (high-level API) that runs on the OS/2 environment.
HLL.  High-level language.

I

ID.  Identifier.

**identifier (ID).** One or more characters used to identify or name data elements and possibly to indicate certain properties of that data element.

immediate notification.  A function available in the notification facility that enables your program administrator to have messages sent to specified users as problems and changes are reported and entered into the database.

**immediate response chain (IRC).** A linked series of panel responses. An IRC is entered during the prompting sequence for immediate one-time running.

**Information/Management.** Information/Management is a program that extends a user’s ability to gather, organize, and locate information. The program provides a structure for organizing a data processing installation’s problem information, change information, and configuration information. The program also provides a means for loading, searching, reviewing, and reporting data. Through its Panel Modification Facility (PMF), Information/Management provides an installation with the means to tailor the Problem Management, Change Management, and Configuration Management functions shipped with Information/Management. PMF also allows an installation to develop its own applications.

**input data.** Data received or to be received by a functional unit or by any part of a functional unit.

inquiry argument.  A definition of data used with inquiry transactions. This can be either structured or freeform responses.

**installation.** (1) In system development, preparing and placing a functional unit in position for use. (2) A particular computing system, including the work it does and the people who manage it, operate it, apply it to problems, service it, and use the results it produces.

**Installation Tailoring Facility.** That portion of the product that eases installation of the product or migration to Information/Management with a series of interfaces that have menus and choices.

**Integration Facility.** An Information/Management application that models a typical change and problem management system and provides interfaces to several products, integrating them into one system to help you initiate your system management tasks.

**Interactive Problem Control System (IPCS).** A component of VM that permits online problem management, interactive problem diagnosis, online debugging for disk-resident CP ABEND dumps, problem tracking, and problem reporting.

**Interactive System Productivity Facility (ISPF).** An IBM licensed program that serves as a full-screen editor and dialog manager. Used for writing application programs, it provides a means of generating standard panels and dialogs between the host computer and terminal user.

**IPCS.** Interactive Problem Control System.

**IRC.** Immediate response chain.

**ISPD TLC.** The ISPF DTL conversion utility.

**ISPF.** Interactive System Productivity Facility.
J

JCL.  Job control language.

JES2.  Job Entry Subsystem 2.


job control language (JCL).  A control language used to identify a job to an operating system and to describe the job’s requirements.

Job Entry Subsystem 2 (JES2).  An MVS subsystem that receives jobs into the system, converts them to internal format, selects them to be run, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing. See also JES3.

Job Entry Subsystem 3 (JES3).  An MVS subsystem that receives jobs into the system, converts them to internal format, selects them to be run, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them through a common job queue. See also JES2.

journal field.  A special Information/Management field identified by the symbol <H> that triggers Information/Management programs to keep a history of changes to a particular record. Whenever data is entered or updated in the field, the date, time, and user making the changes are recorded with the record data. This data is maintained as long as the record is kept in the database.

journalize.  To keep a history of changes to individual fields in a given Information/Management record.

K

keylist.  ISPF defines and stores key assignments active for an application panel in keylists. These key assignments allow the user to request commands and other actions through the use of function keys. Key assignments for an application are displayed in the function key area of application panels.

key phrase.  (1) A short description of a selection taken on an option or selection panel. It consists of two parts—a description and a selection name—separated by an equal sign (for example, RECS=PROBLEM). (2) A short description of a nonfield selection taken from a data-entry panel. It consists of the data following the selection number (for example, REPORTER DATA). See also visible phrase.

key-sequenced data set.  A data set in which the access path is based on the contents of one or more key fields contained in the records.

L

line command.  A command issued in the line command field on a table panel or by using the LINECMD command. A line command performs the action against the record on the specified line.

link-edit.  To create a loadable computer program by means of a linkage editor.
**list item.** A type of response field that is processed in tabular form and allows multiple instances of the same type of data. This format allows skipped entries.

**list processor.** A program exit that enables entry, update, and display of list items.

**LLAPI.** Low-level application program interface (low-level API).

**load module.** A computer program in a form suitable for loading into main storage to start and run; it is usually the output of a linkage editor.

**local shared resources (LSR).** An option for sharing I/O buffers, I/O-related control blocks, and channel programs among VSAM data sets in a resource pool that serves one partition or address space.

**logical database partitioning.** A means of organizing data in the Information/Management database into “logical partitions.”

**logical unit (LU).** A type of network addressable unit that enables end users to communicate with each other and gain access to network resources.

**low-level application program interface (low-level API or LLAPI).** A program interface that enables user-written programs or the HLAPI to access selected Information/Management functions. It provides more user control than does the HLAPI. See also *high-level application program interface.*

**LSR.** Local shared resources.

**LRECL.** Logical record length.

**LU.** Logical unit.

**M**

**Management application.** The portion of the product that enables a user to record and track problems, changes, and configurations.

**manual entry block.** A control block used to store assisted-entry panel responses to verify each item in the response.

**machine readable information (MRI).** Pertaining to data a machine can acquire or interpret (read) from a storage device, a data medium, or other source.

**MEB.** Manual entry block.

**message chain.** A linked list of messages generated by Information/Management when processing a transaction.

**mixed case data.** A data string that can contain uppercase, lowercase, or a combination of uppercase and lowercase SBCS data.

**mixed data.** A character string where portions can consist of SBCS data and other portions can consist of DBCS data. See also *shift in, shift out.*

**model PIDT.** A PIDT whose header rows are used as a model to build a dynamic PIDT. Any PIDT can be used as a model.

**model record.** A type of record that enables you to model software and hardware components. Model records are useful in creating multiple components of the same type (for example, terminals).

**MRI.** Machine readable information.

**MRES.** Multiclient remote environment server.
**multiclient remote environment server (MRES).** An Information/Management server that provides Information/Management API functions to multiple clients concurrently. Characteristics are:

- Started by the MVS operator before a request is received from a client.
- Processes multiple conversations concurrently within a single address space.
- Contains multiple client conversation processors and tasks to manage the client conversation processors.
- Runs until stopped by the MVS operator.
- Connections provided by APPC or TCP/IP.

**multiple pattern group.** A definition associated with assisted-entry panels that indicates multiple prefixes are collected with each response.

**multiple response item.** A definition given to a structured prompting field that accepts more than a single response word at data entry.

**Multiple Virtual Storage (MVS).** An IBM licensed program whose full name is the Operating System/Virtual Storage (OS/VS) with Multiple Virtual Storage/System Product for System/370. It is a software operating system that controls the starting and running of programs.

**Multisystem Database Access.** The facility in Information/Management that enables users on different BLX-SPs to concurrently share Information/Management VSAM data sets. The BLX-SPs that share the data sets can reside either on the same MVS/ESA system or on multiple MVS/ESA systems.

**MVS.** Multiple Virtual Storage.

**MVS/ESA.** MVS/Enterprise Systems Architecture.

**National character.** One of several characters (@, #, and $) of special use in Information/Management.

**National language support (NLS).** The modification or conversion of a product to conform to the requirements of another language or country. This can include the enabling or retrofitting of a product and the translation of nomenclature, machine-readable information, or documentation of a product.

**NCCF.** Network Communications Control Facility.

**NCP.** Network Control Program.

**NetView.** A System/370-based IBM licensed program used to monitor a network, manage it, and diagnose its problems.

**NetView AutoBridge.** A feature of Information/Management that serves as an application enabler for the Information/Management NetView Bridge Adapter. AutoBridge receives data from specific alerts, messages, and other applications through its API and uses the data to build and perform Information/Management transactions.

**NetView Bridge Adapter.** A feature of Information/Management that provides a link to the NetView Bridge function of NetView, which in turn allows remote users in your network to open and update network problems logged in the Information/Management database.

**NetView command list language.** An interpretive language unique to the NetView program that is used to write command lists.
Network Communications Control Facility (NCCF). An IBM licensed program consisting of a base for command processors that can monitor, control, and improve the operation of a network.

Network Control Program (NCP). An IBM licensed program that provides communication controller support for single-domain, multiple-domain, and interconnected network capability.

Network Problem Determination Application. (NPDA) An IBM licensed program that helps the user identify network problems from a central control point using interactive display techniques.

NLS. National Language Support.

nonmultiple pattern group. A definition associated with assisted-entry panels that indicates multiple prefixes are not collected with each response.

not logic (¬). A special character used to specify search arguments in which Boolean not logic (¬) is applied to the data when processing a response field.

notification facility. An Information/Management facility that provides two functions: immediate notification and problem escalation. See also immediate notification and problem escalation.

NPDA. Network Problem Determination Application.

O

OPC/A. Operations Planning and Control/Advanced.


Operations Planning and Control/Advanced (OPC/A). A program that plans, controls, and automates MVS batch production workload.


operator station task (OST). The NetView task that establishes and maintains the online session with the network operator. There is one operator station task for each network operator who logs on to the NetView program.

or logic (|). A special character used to specify search arguments in which Boolean or logic (|) is applied to the data when processing a response field.

OST. Operator station task.

OS/2. IBM Operating System/2.

overview panel. A panel that provides information about the Information/Management program. Subjects include:

- A product overview
- Command descriptions
- Instructions for using the commands
- A summary of what is new in the latest release.

owning partition. In logical database partitioning, the Owning Partition Name is contained within the record and identifies the partition which owns the record.
P

PALT. Program interface alias table.

panel control block. A control block containing information relevant to a panel.

Panel Modification Facility (PMF). A facility of Information/Management that helps users modify and create panels that are more specific to their organization’s needs.

parameter. Data passed to a program or procedure by a user or another program, namely as an operand in a language statement, as an item in a menu, or as a shared data structure.

parent record. In Information/Management, a record that is linked to other (child) records. A parent record can have more than one child record.

partitioned data set (PDS). A data set in direct-access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

PCB. Panel control block.

PDF. Program Development Facility of ISPF.

PDS. Partitioned data set.

PF key. Program function key. See also function key.

PIAT. Program interface argument table.

PICA. Program interface communications area.

PIDT. Program interface data table.

PIHT. Program interface history table.

PIMB. Program interface message block.

PIPT. Program interface pattern table.

PIRT. Program interface results table.

PLC tape. Program level change tape.

PMF. Panel Modification Facility.

PPI. Program-to-program interface.

predefined variable. Built-In values provided by Information/Management to assist you in using the Report Format Facility. There are two types of predefined variables: read-only and read/write. All predefined variables begin with the character Z.

prefix. A keyword used in a search argument that identifies which field in the database the data being searched corresponds to. A prefix also contains a slash (/) or an underscore character (_).

prefix index. A type of dictionary index key that identifies p-words that are in the Information/Management dictionary data set.

prefix word (p-word). Consists of a prefix, prefix index, and validation data. See also prefix.

primary partition. In logical database partitioning, the Primary Partition Name is contained within a privilege class record and identifies the logical partition, and thus the records, to which the privilege class has access.
privilege class. To protect the information in your database from being viewed, altered, or erased by unauthorized users, your organization can set up privilege classes. The privilege classes define which tasks an individual user or group of users can perform.

problem data. Information that is identified, collected, tracked, and analyzed using Problem Management.

problem escalation. A function available in the notification facility that enables the program administrator to set escalation criteria and send alert messages to those you designate and at specified intervals until the problem is resolved.

Problem Management Facility. An online Information/Management facility that helps you document, review, monitor, and report problems with any hardware, software, procedure, or publication at your installation.

program administrator. A person at a user’s site who administers and maintains the Information/Management files, programs, and problems.

program call facility. An authorized program that communicates requests from users’ address spaces to BLX-SP. BLX-SP then instructs VSAM to perform the I/O processing.

program development facility of ISPF. A tool that you use to create and maintain applications and other types of data. It enables you to develop and test applications online, and includes utilities for editing, browsing, and controlling your library.

program function key. On a display device keyboard, a key that passes a signal to a program to call for a particular operation.

program interface alias table (PALT). One of several program structures that support the transactions your application uses to access the Information/Management database. The alias table lets your application specify alias names for PIDTs, p-words, p-word indexes, and s-word indexes. It also enables you to specify default values for fields.

program interface argument table (PIAT). One of several program structures that support the transactions your application uses to access the Information/Management database. The PIAT contains a list of freeform arguments used in an inquiry.

program interface communications area (PICA). One of several program structures that support the transactions your application uses to access the Information/Management database. The PICA is used to communicate between the LLAPI and your application. The PICA also serves as an anchor to all other LLAPI structures.

program interface data table (PIDT). One of several program structures that support the transactions your application uses to access the Information/Management database. The PIDT is a view of a particular type of Information/Management database record. The LLAPI uses these PIDTs when it processes interface transactions.

program interface history table (PIHT). One of several program structures that support the transactions your application uses to access the Information/Management database. The PIHT contains history data. A PIHT is created when a retrieve transaction requests processing of history data.

program interface message block (PIMB). One of several program structures that support the transactions your application uses to access the Information/Management database. The PIMB defines the format of a message block on the message chain.

program interface pattern table (PIPT). One of several program structures that support the transactions your application uses to access the Information/Management database. The PIPT contains the validation criteria used to verify response data in an Information/Management operation.
program interface results table (PIRT). One of several program structures that support the transactions your application uses to access the Information/Management database. The PIRT contains a list of external record IDs found that meet specific search criteria.

program-level change tape (PLC tape). A system update service that includes new functions as well as cumulative system changes. The latest PLC tape contains all new updates, as well as all previous updates since the last release base.

program temporary fix (PTF). A temporary solution or bypass of a problem diagnosed by IBM resulting from a defect in a current unaltered release of the program.

program-to-program interface (PPI). A facility in the NetView program that allows user programs to send data buffers to, or receive data buffers from, other user programs. It also enables system and application programs to send alerts to the NetView hardware monitor.

prompting sequence. An ordered series of panels to which you respond to complete a given task.

pseudoconversational mode. See CICS pseudoconversational mode.

PTF. Program temporary fix.

pull-down menu. In Common User Access architecture, a list of choices extending from a selected action-bar choice that gives users access to actions, routings, and settings related to a panel.

p-word. Prefix word.

Q

QSAM. Queued sequential access method.

queued sequential access method (QSAM). An extended version of the basic sequential access method (BSAM). When QSAM is used, a queue is formed of input data blocks that are awaiting processing or of output data blocks that have been processed and are awaiting transfer to auxiliary storage or to an output device.

quick search. A type of search that guides you with a fill-in-the-blanks prompting sequence. As you fill in the blanks, the search arguments are built for you. This provides a shorter inquiry path, which decreases the length of time you spend creating your search arguments.

R

RAB. Record anchor block.

RACF. Resource Access Control Facility.

RBA. Relative byte address.

RDO. Resource definition (online).

receive database. When using the Automatic Log Save function, a receive database is one that receives the offloaded SDLDS from a send database and creates and stores Information/Management records.

receive data set. When using the Automatic Log Save function, a receive data set is the sequential data set that the send data set is received as.
**record anchor block.** The anchor for the chain of dialogs and the administrative items that make up a record. It is also an anchor for the SDEs.

**record number identifier (RNID).** The external identifier of a record in the Information/Management database.

**relational data mapping table (RDMT).** Output from BLGUT8, this table is a control block that specifies the mapping of Information/Management data into a relational format.

**relative byte address.** The displacement of a data record or a control interval from the beginning of the storage space allocated to the data set or file to which it belongs.

**relative record data set.** A data set in which each record is assigned a record number according to its relative position within the data set storage space.

**remote environment server (RES).** An Information/Management server that provides Information/Management API functions to a single client. The characteristics of a RES are:

- It is initiated in response to a request from a client.
- It processes only one client conversation.
- Each RES has its own address space.
- When the client ends the conversation, the RES closes down.
- The connection is provided by APPC.

**reply transaction.** After a request transaction is processed, a reply can optionally be returned to the requesting enterprise-developed command.

**report exit routine.** A routine that receives control when a specified event occurs, such as an error.

**Report Format Facility.** An Information/Management facility for producing a wide range of reports within the Information/Management environment. The Report Format Facility lets you print individual reports, run standard or customized reports, modify or create new reports, and graph your reports.

**report format table (RFT).** A table that defines an Information/Management report. The RFT consists of statements that define the data that is to appear in the report and the way the data is to be formatted.

**requester.** A transaction program that provides workstation access to the HLAPI through the server.

**request transaction.** This request, issued by an enterprise-developed command, is forwarded by the NetView Bridge Product to a database server that runs the transaction processor requested by the request transaction.

**RES.** Remote environment server.

**Resource Access Control Facility (RACF).** An IBM licensed program that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected access to protected resources.

**resource definition online (RDO).** A CICS interactive facility to create and modify system resources.

**Resource Measurement Facility (RMF).** A function that, in conjunction with System Availability Management, provides the means to track, record, and report software and hardware availability on MVS/XA systems.

**response buffer.** A storage area where response data is stored. See also *data buffer.*
**response data.** The data that is accepted in a response field.

**response field.** A field on a panel that can accept data. The data can be entered directly or as a result of an immediate response chain entered on the command line.

**Restructured Extended Executor Language (REXX).** A general-purpose programming language particularly suitable for EXEC procedures, XEDIT macros, or programs for personal computing.

**return code.** A value returned to a program to indicate the results of an operation requested by that program.

**REXX.** Restructured Extended Executor Language.

**REXX HLAPI/2.** A function that allows an OS/2 REXX program to use HLAPI/2 transactions.

**REXX HLAPI/AIX.** A function that allows an AIX REXX program to use HLAPI/AIX transactions.

**RFF.** Report Format Facility.

**RFT.** Report format table.

**RMF.** Resource Measurement Facility.

**RNID.** Record number identifier.

**rules record.** An Information/Management record type defined specifically for problem escalation.

\[S\]

**SAM.** Systems availability management.

**SBCS.** Single-byte character set.

**SCP.** System control programming.

**SDDS.** Structured description data set.

**SDE.** Structured description entry.

**SDIDS.** Structured description index data set.

**SDLC.** Synchronous Data Link Control.

**SDLDS.** Structured description log data set.

**SDR.** Structured description record.

**search argument.** An independent variable used to find a match.

**secondary partition list.** In logical database partitioning, the Secondary Partition List defines partitions additional to the Primary Partition to which a privilege class has access.

**send database.** A production Information/Management database running with an SDLDS. The send database is periodically offloaded by the Automatic Log Save Facility or the DB2 Extract Facility for data propagation. See also *Automatic Log Save Facility, DB2 Extract Facility, and data propagation.*

**send data set.** The data set created by the Automatic Log Save Send Function. This data set contains the offloaded SDLDS from a send database. The send data set is used for data propagation by the Automatic Log Save Receive Function and the DB2 Extract Utility.
also Automatic Log Save Facility, DB2 Extract Facility, data propagation, and send database.

**sequential data set.** A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape.

**server.** A functional unit that provides shared services; for example, a file server, a print server, a mail server.

BLX-SP is a data server. The RES and MRES are remote environment servers.

An MVS-based transaction program that resides on the host system. It provides the link between Information/Management and the client system.

**server support API.** Refers to the routines provided in the server support API component of NetView to support building of a transaction server.

**Service Level Reporter/Recorder (SLR).** A program used to monitor an entire installation, including host and networks.

**session parameter.** A constant that determines the operating characteristics of an address space.

**session-parameters member.** A data set that contains the constants (session parameters) that determine the operating characteristics of an address space.

**sharing partition list.** In logical database partitioning, the Sharing Partition List permits a record to be shared by multiple partitions by adding additional partition entries to that identified by the Owning Partition Name.

**shift in (SI).** The SI character signifies the end of any DBCS portion of a mixed data field. It consists of the single EBCDIC shift-in byte X'0F'. See also **shift out, mixed data.**

**shift out (SO).** The SO character signifies the start of any DBCS portion of a mixed data field. It consists of the single EBCDIC shift-out byte X'0E'. See also **shift in, mixed data.**

**SI.** Shift in.

**single-byte character set (SBCS).** An SBCS character is the same as an EBCDIC character. It consists of a single byte in the range X'00'-X'FF'. Two SBCS characters, shift out (SO) and shift in (SI), have special meanings within the context of a mixed data field. See also **shift in and shift out.**

**SLR.** Service Level Reporter/Recorder.

**SMP.** System Modification Program.

**SNA.** Systems Network Architecture.

**SO.** Shift out.

**special character.** One of several characters (#, @, $, /, and &) that have a particular use within Information/Management.

**SQL Generator.** The programs that create SQL statements using RDMTs and a send data set to propagate changes in the Information/Management database to DB2. The SQL statements are placed in the SQL data set.

**SQL data set.** The data set containing the SQL statements created by the SQL Generator.

**SRC.** Stored response chain.

**standard panel style.** Displays Information/Management panels as in versions previous to Version 6.1 without action bars and pull-down menus. Programmable function keys are
defined in each user's ISPF profile. See also action bar, pull-down menu, and function key. Contrast with enhanced panel style.

**stored response chain (SRC).** A predefined, named set of panel responses that resides in the database.

**string field.** A response field that can contain multiple words and special characters.

**structured argument.** A response field entered on a prompted sequence quick search panel or a simulation of quick search processing.

**structured description data set (SDDS).** A VSAM key-sequenced data set that contains a collection of data, including SDEs, related to an Information/Management record.

**structured description entry (SDE).** The data area that maps the fields that represent collected information related to responses made by a user to the structured prompting sequence panels.

**structured description index data set (SDIDS).** A VSAM key-sequenced data set that contains an index to the records stored in the SDDS. It is used to speed up the search for records in the SDDS.

**structured description log data set (SDLDS).** A VSAM entry-sequenced data set. It is an optional data set that is used to store copies of the records written to the Information/Management SDDS.

**structured description record (SDR).** A data area that is the general mapping of the format of an SDDS record. The fields defined by this area constitute the header of the SDDS record.

**structured search.** A type of search that uses a prompting sequence to create the required search arguments.

**structured word (s-word).** A keyword that identifies the contextual meaning of a collected item of data in Information/Management.

**s-word.** Structured word.

**s-word index.** A type of dictionary index key that identifies structured words that are in the Information/Management dictionary data set.

**Synchronous Data Link Control (SDLC).** A discipline conforming to subsets of the Advanced Data Communication Control Procedures (ADCCP) of the American National Standards Institute (ANSI) and High-Level Data Link Control (HDLC) of the International Organization for Standardization, for managing synchronous, code-transparent, serial-by-bit information transfer over a link connection. Transmission exchanges may be duplex or half-duplex over switched or nonswitched links. The configuration of the link connection may be point-to-point, multipoint, or loop.

**SYSGEN.** System generation.

**SYSOUT.** System output stream.

**system administrator.** The person at a computer installation who designs, controls, and manages the use of the computer system.

**System application.** The portion of the product that provides access to data in the Information/Management database. This portion of the product enables an installation to set up privilege class records, reference records, Automatic Log Save records, and other records that affect what the user can do while using the Management application.
systems availability management (SAM). A function of Resource Measurement Facility (RMF) that provides the means to track, record, and report software and hardware availability on MVS/XA systems.

system control programming (SCP). IBM-supplied programming that is fundamental to the operation and maintenance of the system. It serves as an interface with licensed programs and user programs and is available without additional charge.

system generation (SYSGEN). The process of selecting optional parts of an operating system and of creating a particular operating system tailored to the requirements of a data processing installation.

System Modification Program (SMP). A program used to install software and software changes on OS/VS1 and OS/VS2 systems.

Systems Network Architecture (SNA). The description of the logical structure, formats, protocols, and operational sequences for transmitting information units through, and controlling the configuration and operation of, networks.

Note: The layered structure of SNA enables the ultimate origins and destinations of information, that is, the end users, to be independent of and unaffected by the specific SNA network services and facilities used for information exchange.

system output stream (SYSOUT). An indicator used in data definition statements to signify that a data set is to be written on a system output unit.

T

TAF. Terminal Access Facility.

task. In the TME 10 environment, the definition of an action that must be routinely performed on various managed nodes throughout the network. A task defines the executables to be run when the task is executed, the authorization role required to execute the task, and the user or group name under which the task will execute.

Terminal Access Facility (TAF). In the NetView program, a facility that enables a network operator to control a number of subsystems. In a full-screen or operator control session, operators can control any combination of such subsystems simultaneously.

terminal simulator communications area (TSCA). A place for communications between Information/Management and a TSP, and between a TSP and a user-written exit routine.

terminal simulator panel (TSP). A panel in Information/Management through which a user can control simulation of terminal input and output.

terminal simulator EXEC (TSX). A REXX EXEC through which a user can control simulation of terminal input and output.

think time. The elapsed time between receipt of a system response at a terminal and the time when a new transaction is entered.

time sharing option (TSO). An option on an operating system that provides interactive time sharing from remote terminals.

TME 10. The brand name for the suite of Tivoli applications that enable system administrators to manage their network computing enterprise according to four management disciplines: availability management, deployment management, operations and administration, and security management.
TME 10 Enterprise Console (TEC). A TME 10 product that collects, processes, and automatically initiates corrective actions for system, application, network, and database events; it is the central control point for events from all sources. The TME 10 Enterprise Console provides a centralized, global view of the network computing environment; it uses distributed event monitors to collect information, a central event server to process information, and distributed event consoles to present information to system administrators.

TME 10 Global Enterprise Manager (GEM). A TME 10 product that includes connection services enabling mainframe networking products to share data with distributed networking products and that allows a system administrator to manage an enterprise's network from a mainframe console or a distributed workstation. GEM includes a Java viewer that allows the system administrator to view the status of business applications as well as networking components.

TME 10 Software Distribution. A TME 10 product that automates software distribution to clients and servers in a network computing environment. An organization can use this product to install and update applications and software in a coordinated, consistent manner across a network. TME 10 Software Distribution creates file packages and distributes them to predefined subscribers.

TP. Transaction program.

transaction. A specific set of input data that triggers a specific processor task to start and run.

transaction control record (XCR). A data area used to map a transaction control record that can exist in either a Version 2.1 or Version 3 structured description database.

transaction processors. Perform activities on the database system with which they interface. They can retrieve, update, add, and delete information from the database system.

transaction program. A program that performs services related to the processing of an exchange of data. See also conversation.

TSCA. Terminal simulator communications area.

TSO. Time sharing option.

TSP. Terminal simulator panel.

TSX. Terminal simulator EXEC.

U

unflatten. To restore data extracted by the FLATTEN control line into a database that is equivalent to the one from which the data was originally obtained.

universal partition access authority. In logical database partitioning, a privilege class can have Universal Partition Access Authority. A user using a privilege class with Universal Partition Access Authority has access to all partitions, and thus all records, in the Information/Management database.

user-defined line command. A line command that is unique to a user's installation and that can be used on various search results lists.

user profile. A set of default values that Information/Management uses each session. The values can be control information, which tells Information/Management how to perform, or data that the user enters repeatedly during an Information/Management session.
**user exit.** A point in an IBM-supplied program at which a user exit routine may be given control.

**user exit routine.** An installation specific, user-defined program that interfaces directly with the terminal simulator communications area (TSCA).

**user’s address space.** The range of addresses available to an Information/Management user, including interactive users and batch jobs.

---

**V**

**validation record.** A record which contains validation criteria for a field in a record.

**virtual storage (VS).** (1) The storage space that may be regarded as addressable main storage by the user of a computer system in which virtual addresses are mapped into real addresses. The size of virtual storage is limited by the addressing scheme or the computer system and by the amount of auxiliary storage available, not by the actual number of main storage locations. (2) Addressable space that is apparent to the user as the instructions and the data are mapped into the processor storage locations.

**virtual storage access method (VSAM).** An access method for indexed or sequential processing of fixed- and variable-length records on direct-access devices. The records in a VSAM data set or file can be organized in logical sequence by means of a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by means of relative-record number.

**visible phrase.** (1) A short description of a selection taken on an option or selection panel. It consists of two parts, a description and a selection name, separated by an equal sign (for example, RECS=PROBLEM). (2) A short description of a nonfield selection taken from a data-entry panel. It consists of the data following the selection number (for example, REPORTER DATA). See also key phrase.

**VS.** Virtual storage.

**VSAM.** Virtual storage access method.

---

**W**

**ward 42 character.** A double-byte character that contains X’42’ in the first byte and has an EBCDIC value in the second byte.

**watermark character.** The first character of an s-word. This character cannot be typed by the user, so it is impossible for the user to enter an s-word on a command line.

**WDS.** Window Display Services.

**Window Display Services (WDS).** The component of Information/Management that provides other applications within the product with a single full-screen interface for displaying or updating tabular data.

---

**X**

**XCR.** Transaction control record.
Special Characters

¬.  See not logic.
|.  See or logic.
Index

A

ABEND disposition field 11-17
LOGSAVE record 11-17, 11-21
accelerator keys 5-7, 5-20
access to propagated data 11-6
DB2 data 11-7
Information/Management data 11-6
activity records 7-20
administration window, illustration 5-5
administrator, program
creating training records B-1
name C-1
phone number C-1
alias record
creating 8-6
use with product commands 9-1
use with user line commands 8-1
APIPRINT for Archiver 15-1, 15-3, 15-5
Archiver
APIPRINT 15-1, 15-3, 15-5
APPLID= 15-1, 15-9
ARCHIVE= 15-1, 15-6
ARCSESS= 15-1, 15-7
ARG= 15-1, 15-8
BLGARCP 15-1, 15-3, 15-5, 15-18
BLGINREC 15-1, 15-5, 15-17
BLGOUT 15-1, 15-3, 15-5
BLGYACI 15-1
BLGYCHI 15-1
BLGYDYN 15-1, 15-8
BLGYPRI 15-1, 15-3
CHDPIDT= 15-1, 15-10
CHECKIO= 15-1, 15-10
CHILD= 15-1, 15-9
CLASS= 15-1, 15-7
COPY= 15-1, 15-6
DYNMPIDT= 15-1, 15-8
FUNCTION= 15-1, 15-6
Archiver (continued)
HISTORY= 15-1, 15-14
HITS= 15-1, 15-10
input parameters for 15-1
INPUT= 15-1, 15-6
INQPIDT= 15-1, 15-8
inquiry PIDT 15-1, 15-3
LIVESESS= 15-1, 15-7
logging 15-1, 15-3, 15-5, 15-8
LOGGING= 15-1, 15-3, 15-5, 15-8
PARENT= 15-1, 15-9
PURGE= 15-1, 15-6, 15-11
RFTDD 15-1, 15-5
sample JCL for 15-1, 15-2
SBLMSAMP 15-1, 15-3
TEXT= 15-1
TEXTAREA= 15-1, 15-13
TEXTAUD= 15-1, 15-13
TEXTCNT= 15-1, 15-13
TEXTWID= 15-1, 15-13
TEXTXFER= 15-1, 15-13
transaction logs 15-1, 15-3
assigning authorities
description 2-11
example for Problem Management users 2-12
assignment authority 2-7
asynchronous data propagation 11-4
audit trail for freeform text 1-15
authority 11-15
assigning 2-11
assignment 2-7
close 2-8
creation 2-27
database administration 6-1
defining 2-1
defining an individual class 2-14
defining group 2-27
delete 2-6
authority (continued)
deleing 2-44
deleing a member 2-42
determing all 2-34
dictionary update 2-9
display 2-6
displaying 2-39
extry 2-6
examples of creating privilege class 2-13
field level 2-7
for LOGSAVE record 11-15
function level 2-6
group class 2-3
group privilege class contact 2-4
individual class 2-3
master class 2-2
PMF 2-8
record ownership 2-13
structure 2-2
structure example 2-4
transfer-to field 2-3
types 2-5
update 2-6
Authority Entry panel 2-18, 2-22
authorization ID 11-21
LOGSAVE record 11-21
Automatic Log Save Facility
See also Automatic Log Save Receive, Automatic Log Save Send
disabling automatic restart 12-33
error recovery 12-35
implementation planning 12-1
JCL members 12-15
performance 11-14
starting, normal 12-32
using with DB2 Extract 11-12
using with DB2 Extract Facility 13-35
Automatic Log Save Receive
determining which records are received 12-32
error recovery 12-35
filtering records 12-32
LOGSAVE record 12-22
overview 12-1
processing steps 12-4
starting 12-32
stopping 12-32
synchronization 12-38
restoring with BLGUT1 12-41
restoring with GDG 12-38
restoring without BLGUT1 12-42
Automatic Log Save Receive (continued)
synchronization (continued)
restoring without GDGs 12-41, 12-42
synchronization, restoring 12-37
task overview 12-21
Automatic Log Save Send
BLGDUMP1 TSP 12-20
error recovery 12-35
LOGSAVE record 12-7, 13-13
overview 12-1
processing steps 12-3
starting 12-32
stopping 12-32
task overview 12-6
B
BLG#PNAM variable 5-6
BLG#PTYP variable 5-6
BLG#TNAM variable 5-7
BLG00010, System Record Entry 2-15, 8-7, 9-3, 12-8
BLG00020, Reference Entry 8-8
BLG0CMDS command table 5-6
BLG0EN10, Primary Options Menu (System Application) 2-14, 6-2
BLG0J010, Privilege Class Text Entry 2-25
BLG0J100, Class Description Entry 2-16
BLG0J200, Problem Authority 2-23
BLG0J210, Authority Entry 2-22
BLG0J211, Authority Entry 2-18
BLG0J212, Authority Entry 2-18
BLG0J220, Change Authority 2-24
BLG0J260, Class Authority 2-19
BLG0J290, SRC Authority 2-20
BLG0JU00, Class Summary 2-17, 2-24
BLG0KEYS keylist 5-6
BLG0PROF member 1-1
BLG0Y100, Database Administration Selection 6-3
BLG0Y200, Check In 6-6
BLG0Z500, Automatic Log Save Data Set Information Entry 11-16, 12-9, 12-27
BLG0Z530, DB2 Extract Information Entry 13-15
BLG1AA01, function line summary control panel 8-3
BLG1TCRL, Checked Out Record List 6-7
BLG1TDES, Description Text 2-25
BLG6AUDB, Database Administration Authority Level 2-21
BLG6AUUA, Universal Partition Access Authority
Level 4-11
BLGARCP DD for Archiver 15-1, 15-2, 15-3, 15-5, 15-18
BLGDUMP1 TSP 12-20
BLGESADD, increment counter user exit 7-66
BLGESCCL, escalation cleanup user exit 7-67
BLGESCKE, check escalation user exit 7-67
BLGESCLR, clear control block user exit 7-68
BLGESDAT, date and time user exit 7-68
BLGESDUR, duration user exit 7-69
BLGESFCB, free control block user exit 7-70
BLGESGCB, get control block storage user exit 7-70
BLGESGET, get control block field user exit 7-71
BLGESINI, initialize user exit 7-72
BLGESLVL, level increment user exit 7-73
BLGESNOT, notify user exit 7-73
BLGESPRI, priority update user exit 7-75
BLGESPRI, put TSCA data in control block user exit 7-76
BLGESPUV, put variable data user exit 7-77
BLGESSCT, store criteria user exit 7-79
BLGESSEA, get escalation criteria user exit 7-80
BLGINREC for Archiver 15-1, 15-5, 15-17
BLGIRECL variable 1-14
BLGISPFA, ISPF panel 5-5
BLGISPFD, ISPF panel 5-2
BLGISPFE, ISPF panel 5-3
BLGISPFI, ISPF panel 5-4
BLGITEX variable 1-14
BLGLCMDU, Command Alias/Authorization Entry 9-5
BLGLUSER, Users to Notify 7-15
BLGNNSYAL, allocate data set to SYSOUT 7-81
BLGNNSYFR, free data set allocated by BLGNNSYAL 7-82
BLGOUT for Archiver 15-1, 15-3, 15-5
BLGPVARS, User Defined Profile Variables panel 1-4
BLGTDBX1 TSP 13-38
BLGTDBXM TSP 13-29
BLGTDFLT TSP 8-13
BLGTENDL TSP 8-3
BLGTENDS TSP 8-3
BLGTTPSET TSP 1-12
BLGTXEVE DD allocation 7-4
BLGTXEQ TSP 7-7, 7-9, 7-10
BLGTXNOT TSP 7-4
BLGUSERS, extract mail address from USERS record 7-83
BLGUT3EX, recovery user exit 14-1
BLGUT3WT, initialize for receive user exit 14-3
BLGUT4EX, offload a recovery data set user exit 14-6
BLGUT4WT, initialize for send user exit 14-8
BLGUT8, table build utility 13-5
See also table build utility, BLGUT8
BLGYACI for Archiver 15-1
BLGYCHI for Archiver 15-1
BLGYPRI for Archiver 15-1, 15-3
BLMSSGEN, SQL setup, extract, and cleanup user exit 14-11
block size field 11-17
LOGSAVE record 11-17, 11-20
BTNXCN07 TSX 7-4
BTNXPN03 TSX 7-4
build utility BLGUT8 13-5
See also table build utility, BLGUT8

C
Change Authority panel 2-24
changing
profile fields 1-3
changing the notification message 7-28
CHECK IN command 6-1
Check In panel 6-6
Checked Out Record List 6-7
Class Authority panel 2-19
Class Description Entry panel 2-16
Class Summary panel 2-17, 2-24
class, privilege
assigning authorities 2-11
assignment authority 2-7
close authority 2-8
creation 2-27
defining 2-1
defining an individual class 2-14
defining group 2-27
delete authority 2-6
deleting 2-44
deleting a member 2-42
determining all 2-34
dictionary update authority 2-9
display authority 2-6
displaying 2-39
entry authority 2-6
elements of creating privilege class 2-13
class, privilege (continued)
  field level authorities  2-7
  function level authority  2-6
  group class  2-3
  group privilege class contact  2-4
  individual class  2-3
  master class  2-2
  PMF authorities  2-8
  record ownership  2-13
  structure  2-2
  structure example  2-4
  transfer-to field  2-3
  types of authority  2-5
  update authority  2-6
CLIST used to start Information/Management  C-1
  close authority  2-8
  command aliases and authorization  9-1
  COMMAND record  9-1
  context-specific function keys  5-17
  convention
    data-entry  C-2
  copy authority, PMF panel  2-9
  create authority  2-6
    See also entry authority
  creating
    default profiles  1-1
    rules records  7-53

D
  data entry, simplifying  10-1
  data propagation  11-1
    choosing data  11-5
  data integrity  11-5
  enqueues  11-14
  facilities  11-7, 11-9
  frequency  11-5
  locks  11-14
  mode  11-4
  to DB2  11-3
  to Information/Management  11-2
  to Information/Management and DB2  11-4, 13-35
  data set name field  11-16
  LOGSAVE record  11-16, 11-20
  database administration
    CHECK IN  6-1
    checked out record list  6-4
    DBCLEANUP  6-1
    performing CHECK IN  6-4
    performing DBCLEANUP  6-2
  Database Administration Authority Level panel  2-21
  Database Administration Selection  6-3
  database partitioning  4-1
  database type field  11-16
    LOGSAVE record  11-16
  DB2 Extract Facility
    BLGTDX1 TSP  13-38
    BLGTDXM TSP  13-29
    components  11-11, 11-13
    disabling automatic restart  13-34
    error recovery  13-38
    implementation planning  13-1
    JCL members  13-23
    performance  11-14
    processing steps  13-2, 13-4
    starting  13-31
    stopping  13-34
    synchronization, establishing  13-19
    synchronization, restoring  13-38
    using  13-5
    using with Automatic Log Save  13-35
    using with Automatic Log Save Facility  11-12
  DBADMIN authority  2-10
  DBCLEANUP command  6-1
  default profiles
    controlling  1-2
    creating default profiles  1-1
  defining
    escalation criteria  7-53
    group privilege class  2-27
    profile defaults  1-2
  delete authority
    for records  2-6
    PMF panels  2-9
  deleting
    local procedure  C-4
    members of a privilege class  2-42
    privilege class record  2-44
    profile fields  1-4
    SRCs  10-3
  Description Text panel  2-25
  dictionary display authority  2-9
  dictionary update authority  2-9
  disable immediate notification  7-47
  display authority  2-6
  displaying records
    privilege class record  2-39
### E
- **e-mail notification** 7-1
  - editor
  - invocation macro 1-14
  - using PDF 1-14
- enabling immediate notification 7-19
- enhanced panel style
  - accelerator keys 5-7, 5-20
  - action, adding 5-9
  - action, using DTL to add 5-17
  - action, using DTL to change 5-17
  - additional windows, creating 5-21
  - administration window 5-5
- automatic selection 5-6
- BLGISPFA 5-5
- BLGISPFE 5-3
- BLGISPFI 5-4
- description 5-1
- enhanced panel style 5-1
- exploiting ISPF V4 5-7
- global default, changing 5-21
- GUI, enabling 5-7
- inquiry window 5-4
- primary window 5-3
- pull-down menu choice, adding 5-11
- separator bars 5-7, 5-20
- standard panel style 5-1
- text in action bar, changing 5-15
- text in pull-down menu, changing 5-15
- using DTL to change 5-16
- variables for selection 5-6
- entry authority 2-6
- EXEC used to start Information/Management C-1

### F
- field level authorities
  - assignment authority 2-7
  - close authority 2-8
- field validation, changing profile fields 1-3
- freeform text 1-14, 1-15
- function authority for rules 2-6
- function level authority
  - delete 2-6
  - display 2-6
  - entry 2-6
  - update 2-6
- function line summary control panel, BLG1AA01 8-3

### G
- global database partition 4-13
- graphical user interface (GUI)
  - enabling enhanced panel style 5-7
- group privilege class
  - assuming record ownership 2-3
  - group contact 2-4
- GUI (graphical user interface) 5-7
  - See also graphical user interface (GUI)

### I
- identifying records C-2
  - See also numbering records
- immediate notification
  - changing the message 7-9
  - changing the message with TSPs 7-28
  - description 7-1
  - disabling 7-47
  - e-mail with 7-1
  - enabling 7-19
  - example of changing the message with TSPs 7-28
  - queued processing with 7-6
  - TCP/IP SMTP with 7-1
  - TSO send 7-1
  - TSPs 7-18
  - TSPs and TSXs 7-1
- implementing problem escalation 7-50
- individual privilege class, assuming record ownership 2-3
- inquiry PIDT for Archiver 15-1, 15-3
- inquiry window, illustration 5-4
- Integration Facility 7-4
- ISPF (Interactive System Productivity Facility) 1-1
  - See also ?
- ISPF GUI 5-7
  - See also graphical user interface (GUI)
- ISPF panel
  - See also enhanced panel style
  - action bar, changing 5-15
  - action, adding 5-9
  - action, using DTL to add 5-17
  - action, using DTL to change 5-17
  - BLGISPFA 5-5
  - BLGISPFE 5-3
  - BLGISPFI 5-4
  - code, changing 5-8
ISPF panel (continued)
  creating 5-21
  default, changing 5-8
  enhanced panel style 5-3
  global default, changing 5-21
  individual default, changing 5-21
    permanent 5-21
    temporary 5-17
  keylists, changing 5-17
  pull-down menu choice, adding 5-11
  pull-down menu choice, changing 5-15
  using DTL to change 5-16
  where reside 5-5

ISPF tables
  BLG0CMDS 5-6
  BLG0KEYS 5-6
  used by enhanced panel style windows 5-6
  where reside 5-6

K
  keylist, ISPF 5-17
    See also ISPF panel

L
  LAST_SDLDS_KEY 12-38
    See also sequence number
  logical database partitions 4-1
  logical record length field 11-16
    LOGSAVE record 11-16, 11-20
  LOGSAVE record 11-15
    ABEND disposition field 11-17, 11-21
    authority 11-15
    authorization ID 11-21
    block size field 11-17, 11-20
    data set name field 11-16, 11-20
    database type field 11-16
    disabling 12-32, 13-34
    for Send function 12-7, 13-13
    logic 11-15
    logical record length field 11-16, 11-20
    primary allocation field 11-17, 11-20
    receive filter data field 11-19
    receive filter prefix field 11-18
    receive, deleting 12-35
    secondary allocation field 11-17, 11-21
    send, deleting 12-34
    status field 11-19
    synchronize field 11-18

LOGSAVE record (continued)
  time interval field 11-17
  unit type field 11-17, 11-21

M
  mapping records to DB2 13-5
  master privilege class 2-2
  message, immediate notification 7-28
  mnemonic choice selection 5-7
  modifying
    SRC 10-3

N
  naming records C-2
    See also numbering records
  notification
    e-mail notification 7-1
    TCP/IP SMTP with 7-1
    TSO Send 7-1
    TSPs 7-1
    TSXs 7-1
  numbering records
    local procedure C-2

P
  panel
    BLG00010 8-7
    BLG00010, System Record Entry 9-3, 12-8
    BLG0EN10 8-6
    BLG0EN20, Primary Options Menu 7-55
    BLG0Z100, Escalation Rules Entry 7-56
    BLG0Z500, Automatic Log Save Data Set Information Entry 11-16, 12-9, 12-27
    BLG0Z530, DB2 Extract Information Entry 13-15
    BLGI1TVID 7-61
    BLGLALIS 8-9
    BLGPVARS, User Defined Profile Variables 1-4
  training records displayed B-1
  panel copy authority, PMF 2-9
  panel delete authority, PMF 2-9
  panel list authority, PMF 2-9
  Panel Modification Facility (PMF)
    panel list authority 2-9
  panel style, enhanced
    accelerator keys 5-7, 5-20
    action, adding 5-9
    action, using DTL to add 5-17
    action, using DTL to change 5-17
panel style, enhanced (continued)
   additional windows, creating 5-21
   administration window 5-5
   automatic selection 5-6
   BLGISPFA 5-5
   BLGISPFE 5-3
   BLGISPFI 5-4
   description 5-1
   enhanced panel style 5-1
   exploiting ISPF V4 5-7
   global default, changing 5-21
   GUI, enabling 5-7
   inquiry window 5-4
   primary window 5-3
   pull-down menu choice, adding 5-11
   separator bars 5-7, 5-20
   standard panel style 5-1
   text in action bar, changing 5-15
   text in pull-down menu, changing 5-15
   using DTL to change 5-16
   variables for selection 5-6
partitioning databases 4-1
PDF
   invocation macro 1-14
   LRECL 1-14
   using as editor 1-14
performing CHECK IN 6-4
performing DBCLEANUP 6-2
phone number, program administrator’s C-1
PMF authority
   dictionary update 2-9
   PMF panel copy 2-9
   PMF panel delete 2-9
   PMF panel list 2-9
   PMF reports 2-9
   PMF update 2-8
predefined validation patterns 4-20
primary allocation field 11-17
   LOGSAVE record 11-17, 11-20
Primary Options Menu, System Application 6-2
primary window, illustration 5-3
privilege class
   assigning authorities 2-11
   assigning authorities by task 2-11
   assignment authority 2-7
   close authority 2-8
   creation 2-27
   defining 2-1
   defining an individual class 2-14
   privilege class (continued)
   defining group 2-27
   delete authority 2-6
   deleting 2-44
   deleting a member 2-42
   determining all 2-34
   dictionary update authority 2-9
   display authority 2-6
   displaying 2-39
   entry authority 2-6
   examples of creating privilege class 2-13
   field level authorities 2-7
   function level authority 2-6
   group class 2-3
   group privilege class contact 2-4
   individual class 2-3
   master class 2-2
   PMF authorities 2-8
   record ownership 2-13
   structure 2-2
   structure example 2-4
   transfer-to field 2-3
   types of authority 2-5
   update authority 2-6
   Privilege Class Text Entry panel 2-25
   Problem Authority panel 2-23
   problem escalation TSPs 7-52
   PROFILE command
      when changing an SRC 10-3
   profiles
      adding fields 1-4
      BLGPVARS panel 1-4
      changing fields 1-3
      defining a default 1-2
      defining fields 1-4
      defining variables 1-4
      deleting field 1-4
   program administrator
      creating training records B-1
      name C-1
      phone number C-1
   push button
      enabling 5-7
R
RDMT 13-5
See also relational data mapping tables
receive filter data field  11-19
LOGSAVE record  11-19
receive filter prefix field  11-18
LOGSAVE record  11-18
Receive, Automatic Log Save  12-6
See also Automatic Log Save Receive
records
  assigning IDs  C-2
  create alias record  8-6
  identifiers  C-2
  local delete procedure  C-4
  local numbering procedure  C-2
  numbering  C-2
  ownership for security  2-13
  ownership, assuming with privilege class  2-3
  training records, creating  B-1
  transfer-to field  2-3
relational data mapping tables
  BLGUT8  13-5
    building  13-5
    listing  13-16
    sample output  13-11
    table build utility  13-5
REPORT command
  when changing an SRC  10-3
reports authority, PMF  2-9
rules and function authority  2-6
rules record type
  on BLG0Q213 (Authority Summary panel)  2-41
  on panel BLG0J210 (Authority Entry panel)  2-23
rules records
  creating  7-53
  escalation rules entry panel  7-56
S
sample JCL for problem escalation  7-60
SBLMSAMP for Archiver  15-1, 15-3
SBLMTSX data set  7-5
SDLDS (structured description log data set)
  automatic offloading  12-33
search records, determining privilege classes  2-34
searching in problem escalation  7-56
secondary allocation field  11-17
LOGSAVE record  11-17, 11-21
Send, Automatic Log Save  12-6
See also Automatic Log Save Send
separator bars  5-7, 5-20
sequence number
  determining  11-22
  how incremented  11-22
  LAST(SDLDS_SENT)  11-22
  where stored  11-22
setting up SRCs  10-1
sharing partition list  4-13
SMTP (simple mail transfer protocol)  7-7
SRC (stored response chain)
  assessing need for  10-1
  controlling  10-4
  deleting  10-3
  modifying  10-3
  planning  10-1
  privileges of master class  10-4
SRC Authority panel  2-20
standard panel style
  description  5-1
  illustration  5-2
starting escalation processing  7-60
status field, LOGSAVE record  11-19
SYNCH_NUMBER  12-38
See also sequence number
SYNCH_TABLE  12-38
See also sequence number
synchronization  12-1, 13-19
See also Automatic Log Save Receive
See also DB2 Extract Facility
synchronize field, LOGSAVE record  11-18
System Record Entry panel  2-15
T
table build utility, BLGUT8
  RDMT build sample output  13-11
  RDMT build job stream example  13-11
  relational data mapping tables (RDMTs)  13-5
  TABLE/ETABLE statement  13-6
tailoring the escalation facility  7-61
TEXTAUD keyword  1-15
time interval field  11-17
  LOGSAVE record  11-17
training records B-1
  creating B-1
  example displays B-1
transaction logs for Archiver  15-1, 15-3
transfer-to  2-3
transfer-to fields  2-3
TSO Send with notification 7-2
TSPs used for immediate notification
- BLGESC01 7-52
- BLGESC02 7-52
- BLGESC03 7-52
- BLGESC04 7-52, 7-62
- BLGESC05 7-52
- BLGESC06 7-52
- BLGESC07 7-52, 7-63
- BLGESCAL 7-52, 7-60
- BLGNOTAE 7-20, 7-28
- BLGNOTAU 7-20, 7-28
- BLGNOTCE 7-19, 7-28
- BLGNOTCU 7-19, 7-28
- BLGNOTPE 7-19, 7-28
- BLGNOTPU 7-19, 7-28
- BLGTSPAE 7-20, 7-28
- BLGTSPAU 7-20, 7-28
- BLGTSPCE 7-19, 7-28
- BLGTSPCU 7-19, 7-28
- BLGTSPPE 7-19, 7-28
- BLGTSPPU 7-19, 7-28
- control panels 7-28
  - immediate notification message 7-28
TSXs used with e-mail notification 7-1

U
unavailable pull-down choice
- identifying 5-7
unit type field 11-17
  - LOGSAVE record 11-17, 11-21
Universal Partition Access Authority Level 4-11
update authority 2-6
update authority, PMF 2-9
update dictionary authority 2-9
updating
  - profiles, all 1-12
User Defined Profile Variables panel,
  - BLGPVARS 1-4
user exit
  - automatic log save 14-1
  - BLGUT3EX return codes 14-2
  - BLGUT3WT return codes 14-3
  - BLGUT4EX return codes 14-7
  - BLGUT4WT return codes 14-8
  - DB2 extract 14-1
user exit, notification management
  - BLGESADD 7-66
  - BLGESCCL 7-67
user exit, notification management (continued)
  - BLGESCKE 7-67
  - BLGESCLR 7-68
  - BLGESDAT 7-68
  - BLGESDUR 7-69
  - BLGESFCB 7-70
  - BLGESGCB 7-70
  - BLGESGET 7-71
  - BLGESINI 7-72
  - BLGESLVL 7-73
  - BLGESNOT 7-73
  - BLGESPRI 7-75
  - BLGESPUV 7-77
  - BLGESSCT 7-79
  - BLGESSEA 7-80
  - BLGNSYAL 7-81
  - BLGNSYFR 7-82
  - BLGUSERS 7-83
  - BLGUT3EX 14-1
  - BLGUT3WT 14-3
  - BLGUT4EX 14-6
  - BLGUT4WT 14-8
  - BLMSSGEN 14-11
escalation facility 7-65
  - immediate notification 7-65
user ID, MVS 2-1
user line command
  - alias record 8-1
  - BLGTDFLT TSP 8-13
  - control panel BLG1AA01 8-3
  - control panel numbering 8-4
  - data collected by line command processing 8-10
default TSP 8-13
description 8-1
  - identifying collected data 8-5
  - Search List Buffer (SLB) 8-11
  - Search Row Buffer (SRB) 8-11
  - Search User Block (SUB) 8-10
  - SLB (Search List Buffer) 8-11
  - SRB (Search Row Buffer) 8-11
  - steps for defining 8-2
  - SUB (Search User Block) 8-10
  - TSP alias names 8-5
  - TSP for processing 8-10
USERS record
  - alert message 7-60
  - control block (ESCB) 7-65
  - creating 7-10, 7-22
  - creating JCL 7-60
USERS record (continued)
  creating the USERS record 7-22
  description 7-49
  level field 7-57
  process overview 7-50
  search argument 7-56
  starting 7-60
  tailoring the facility 7-61
  TSPs 7-52
using immediate notification 7-1, 7-18
using problem escalation
  escalation command template 7-62
  escalation criteria 7-62
  escalation message 7-63
  overview 7-49
utility BLGUT8 13-5
  See also table build utility, BLGUT8

V
variable pool 1-1
variable used for window selection
  BLG#PNAM 5-6
  BLG#PTYP 5-6
  BLG#TNAM 5-7