# Installing Information/Management Version 6 Release 1 for Client/Server

Document Number GG24-4323-00

July 1994

International Technical Support Organization
Raleigh Center

**Take Note!**

Before using this information and the product it supports, be sure to read the general information under "Special Notices" on page xi.

**First Edition (July 1994)**

This edition applies to Version 6, Release Number 1 of Information/Management for use with MVS/ESA.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader′s feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 545 Building 657
P.O. Box 12195
Research Triangle Park, NC 27709-2195

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

# Abstract

This document describes how to install Information/Management Version 6 Release 1 base code in a Client/Server environment.

It should be used by base software and Information/Management specialists who need to understand the tasks required to implement Information/Management Version 6 Release 1 in an heterogenous environment.

It provides:

APPC/MVS, TCP/IP and CICS/ESA. step by step customization information, required to understand how to integrate all components in a Client/Server environment.

ISPF V4 R1 customization information for using the GUI interface.

Detailed information about availability enhancements in this release.

Housekeeping techniques that are useful for the Information/Management support specialist.

(130 pages)

# Contents

# Figures

# Special Notices

This publication is intended to help systems engineers and systems programmers who need to install and maintain the product. The information in this publication is not intended as the specification of any programming interfaces that are provided by Information/Management. See the PUBLICATIONS section of the IBM Programming Announcement for Information/Management, for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 208 Harbor Drive, Stamford, CT 06904 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

You can reproduce a page in this document as a transparency, if that page has the copyright notice on it. The copyright notice must appear on each page being reproduced.

The following terms, which are denoted by an asterisk (*) in this publication, are trademarks of the International Business Machines Corporation in the United States and/or other countries:

C Set ++
C/2
CICS
CICS/ESA
Common User Access
CUA

DATABASE 2
DB2
Enterprise Systems Connection Architecture
ESCON
GDDM
IBM
Information Warehouse
ISSC
MVS/DFP
MVS/ESA
NetView
OPC
Operating System/2
OS/2
PS/2
RACF
SMP/E
SystemView
VTAM

The following terms, which are denoted by a double asterisk (**) in this publication, are trademarks of other companies:

# Preface

This document describes how to install Information/Management Version 6 Release 1 base code in a Client/Server environment.

It should be used by base software and Information/Management specialists who need to understand the tasks required to implement Information/Management Version 6 Release 1 in an heterogenous environment.

## How This Document Is Organized

The document is organized as follows:

- Chapter 1, "Introduction"

  This chapter gives an overview of the new features and highlights Information/Management′s position within SystemView.

- Chapter 2, "Pre-Installation Tasks"

  This chapter documents the hardware and software requirements and provides planning information for the Information/Management Version 6 Release 1 installation process.

- Chapter 3, "Installation"

  This chapter explains how to install a prime Information/Management Version 6 Release 1 system that can be used as a base for migration.

- Chapter 4, "Migration from Previous Versions"

  This chapter describes the migration tasks from previous versions of Information/Management to Version 6 Release 1.

- Chapter 5, "Installing the Client/Server Interfaces"

  This chapter describes installation and customization steps required in the mainframe to provide support for the new Client/Server interfaces available with Version 6 Release 1.

- Chapter 6, "Multisystem Database Access"

  This chapter explains how to implement access to a single Information/Management database from multiple Information/Management systems.

- Chapter 7, "Setting Up the CICS/ESA Interface"

  This chapter explains how to access Information/Management databases and programs from a CICS/ESA transaction.

- Chapter 8, "Increasing Availability"

  This chapter illustrates how to obtain a higher Information/Management availability by presenting hot standby and data mirroring scenarios based in the Automatic Log Save and DB2 Extract Facility functions.

- Chapter 9, "Maintenance"

  This chapter discusses maintenance and housekeeping techniques for Information/Management Version 6 Release 1.

## Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *Information/Management Planning and Installation Guide*, SC34-4455
- *Information/Management Program Administration Guide and Reference*, SC34-4341
- *Information/Management Operations and Maintenance Reference*, SC34-4480
- *Information/Management Client Installation and User′s Guide*, SC34-4487
- *Application Programming Interface (API) Guide and Reference*, SC34-4463
- *Information/Management Diagnosis Guide*, SC34-4460
- *CICS/ESA Systems Definition Guide*, SC33-0664
- *CICS/ESA Resource Definition (Online)*, SC33-0666

## International Technical Support Organization Publications

The publications listed in this section are redbooks considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *Boosting Your Information/Management Version 6 Release 1 Performance*, GG24-4324 (available August 1994)
- *Jumping out of the Box: New Interfaces and Features of Information/Management Version 6 Release 1*, GG24-4325 (available August 1994)
- *Information/Management Version 5 - Installation and Performance, GG24-3796*

A complete list of International Technical Support Organization publications, with a brief description of each, may be found in:

> *Bibliography of International Technical Support Organization Technical Bulletins,* GG24-3070.

To get listings of redbooks online, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

**How to Order Redbooks**

IBM employees may order redbooks and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-284-4721. Visa and Master Cards are accepted. Outside the USA, customers should contact their IBM branch office.

You may order individual books, CD-ROM collections, or customized sets, called GBOFs, which relate to specific functions of interest to you.

## Acknowledgments

Project manager and author of this document:

Sergio Juri
International Technical Support Organization, Raleigh Center

The authors of this document are:

Mariana Boshoff
ISM South Africa

Arthur Campbell
ISSC United Kingdom

Charles Ngidi
ISM South Africa

Gildas Porcheron
IBM France

This publication is the result of a residency conducted at the International Technical Support Organization, Raleigh Center.

Thanks to the following people for the advice and guidance provided in the production of this document and the time they took to review it.

Bill Couchon
Keith Dean
Bruce Hanover
Nancy Leavell
Clayton Sims
Dave Wright
IBM Information/Management Team, Cary Development Lab

George Canfield
IBM ISPF Team, Cary Development Lab

Carolyn Dean
Ava Vanyo
IBM Information Development, Cary

Carla Sadtler
International Technical Support Organization, Raleigh Center

Chris Shaw
IBM United Kingdom

# Chapter 1. Introduction

Information/Management Version 6 Release 1 introduces some significant changes from previous versions, each of which will be discussed at length in this publication.

In September 1990, IBM* announced SystemView* as the strategy for delivering enterprise-wide systems management solutions. Many of the new features that are being introduced with this version of Information/Management are conforming to that strategy, as the Information/Management product is positioned for the future.

## 1.1 Overview of Information/Management

Information/Management is a product designed to support the processes for handling problem, change, and configuration management. The product can be viewed as having the following base components:

- The BLM and BLG components (formerly separate products, Information/Management and Information/System) together process the data in the various VSAM data sets and control how they interact with each other. This includes any activity that involves I/O to these data sets.
- The BLX component provides supervisor and data management services. This includes virtual storage management, enqueue management, and the opening and closing of data sets.

## 1.2 Evolution of the Information/Management Product

The Information/Management product began in 1978 as two separate products, Information/Management and Information/System. At the time, a user would have been connected to the mainframe with a 3270 terminal (now programmable workstations are widely available, creating new connectivity options). This initial version provided a small subset of the functions that are now available. Tailoring was much more limited, and what tailoring could be achieved was cumbersome.

When Version 2 was released, the major enhancement to the product was the introduction of the Panel Modification Facility (PMF), which has become an integral part of the product. This provided the capability to add, modify, and delete fields on panels. A further significant enhancement with this version was the Report Format Facility, giving the ability to produce reports, which would assist in managing problem and change processes.

With Version 3, enhancements to the configuration management part of the product, and new features such as the Terminal Simulator Panel (TSP), were added. TSPs were to allow a mass batch update capability for the first time.

With Version 4, the major new feature was the application programming interface, API, which extended the way that the product could be used. The Integration Facility provided a quick starting point for new users.

Before Information/Management Version 5, each TSO user had his or her own set of buffers as can be seen in Figure 1 on page 2. Database integrity was ensured through a control record on the database. Each

Information/Management user performing an update to the database would read the control record to check if someone has exclusive access; if not, it could book the control record and proceed.

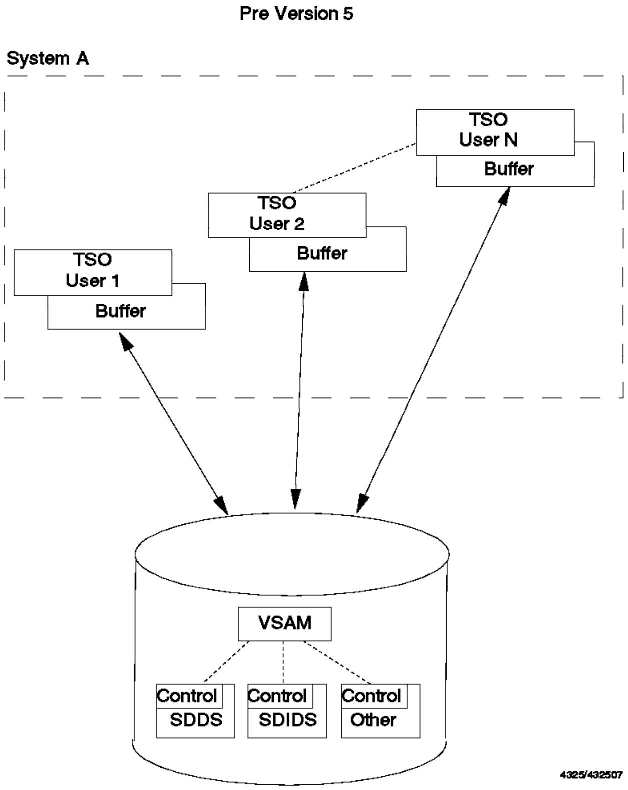


*Figure 1. Information/Management before Version 5*

With Version 5.1, Information/Management and Information/System were merged into one product, and a service provider address space was introduced.

Instead of each user having his or her own buffers, the service provider address space has VSAM buffers allocated to it as seen in Figure 2 on page 3. The service provider address space handles I/O synchronization. There′s no longer need to read the control record, the service provider address space guarantees database integrity .

Information/Management Version 5.1, which was also the first release of Information/Management to exploit MVS/ESA, introduced other new features such as the REXX API, a substantial increase in the database size limit, and the option to edit freeform text with the PDF editor.

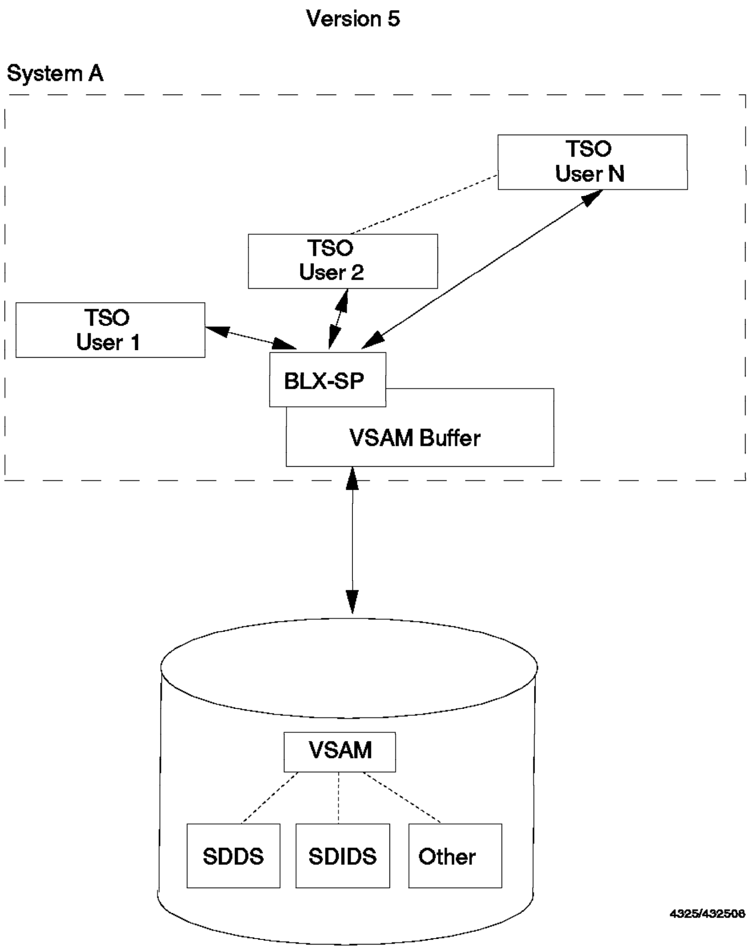


*Figure 2. Information/Management Version 5*

## 1.3 New Features of Version 6 Release 1 of Information/Management

Major new features are available with Information/Management Version 6 Release 1, as well as many enhancements to previously existing product functions. This chapter summarizes these enhancements. For more information, see the Information/Management *Planning and Installation Guide and Reference*, SC34-4455.

**Database Structure Enhancements**

Multisystem Database Access
Users on different MVS/ESA systems can now share Information/Management VSAM data sets without compromising data integrity.

SDIDS 16- and 32-Byte Key Support
Users now have a choice of using either a 16- or a 32-byte key length for the SDIDS. You should use 32-byte keys if you have DBCS. You can also use 32-byte keys with SBCS data to extend your search arguments length to 32 bytes.

Automatic Log Save
This new facility enables you to propagate your production data to another Information/Management database. It could also be used to periodically offload the SDLDS without propagating the data.

DB2* Extract
This new facility enables you to propagate your production data to a DB2 database.

### Installation Enhancements

Installation Tailoring Facility

The Installation Tailoring Facility is a set of dialogs designed to lead the inexperienced installer of Information/Management through installation and customization of Information/Management. The online introduction to the Installation Tailoring Facility can also be useful in providing the experienced installer with detailed information about the Information/Management parameters.

### BLX-SP Enhancements

Multiple BLX-SPs

Information/Management can now run more than one BLX Service Provider (BLX-SP) on an MVS/ESA system at the same time. For example, with this enhancement you can run both a test and a production BLX-SP on the same MVS/ESA* system.

Broadcast Message

The presentation of broadcast messages is enhanced so that broadcast messages appear on a separate panel. This version of Information/Management also includes a time stamp on the messages so that users know when a message was sent. You can also select a single VSAM data set and send a broadcast message to its users.

Blocking Users

By using the UTIL operand with the REALLOC operator command, you can prevent users from accessing the BLX-SP so you can run Information/Management utilities without interruption.

SHUTDOWNWT

When the operator issues the STOP command for a BLX-SP, the BLX-SP stops after SHUTDOWNWT time expires or when no users are connected, whichever comes first.

Invocation ABEND Message

If you attempt to sign on to Information/Management and the BLX-SP is not running, a customizable message explaining the problem is shown before the ABEND occurs. ISPF is not terminated by the ABEND.

### Client Environment

OS/2* and CICS/ESA* Clients

With Information/Management Version 6 Release 1, you can access functions from remote environment workstations on OS/2 or CICS/ESA. This is done utilizing the High Level Application Program Interface (HLAPI), the Remote Environment Server (RES) (RES), and Advanced Program-to-Program Communication (APPC) technology.

### API Enhancements

Enqueue Retry Capabilities

The Low Level Application Programming Interface (LLAPI) now provides automatic retry support for all interfaces. A function can gain access to a record as soon as it is freed by the previous user.

Process History Data

The LLAPI enables you to retrieve history data on the retrieve transaction, modify it or mark it for deletion, and then include it with the create or update transactions.

### Command Enhancements

The RECALL command has been enhanced and the following three new commands have been added: FREE, TRACE, and WINDOW.

### Base Product Enhancements

Panel Styles

With Information/Management Version 6, you can now choose among three different presentation styles: standard, enhanced, or graphical. The enhanced panels are similar in look to a CUA* interface and include action bars and pull-down menus, even if they are not yet fully CUA compliant.

**Base Product Enhancements**

| | |
|---|---|
| The data set reference by the output description name field on the print panel, allows you to write the output to a remote processor. | |
| CSECT Modules | FMID and PTF numbers are part of the prologs of the Information/Management CSECT modules. |

**Improved Documentation**

| | |
|---|---|
| Manuals have been reworked. | The new *Client Installation and User's Guide* describes how to install and implement the new OS/2 and CICS/ESA environments. The previous Information/Management library was also changed to include the *Operation and Maintenance Reference* and the *Program Administration Guide and Reference*. The *Quick Reference* and *User's Guide* are now expanded for our end users. The *Planning and Installation Guide and Reference* was reorganized to follow the flow of the Installation and Tailoring Facility. |

## 1.4 Positioning of the Information/Management Product

With the announcement of SystemView in September 1990, the system management environment as we used to know it, took a significant step. For the first time customers had a system management strategy to manage enterprise-wide information systems. This strategy consists of the SystemView structure and SystemView conforming products. To see how Information/Management fits in this structure see Figure 3 on page 7.

The SystemView structure consists of three dimensions that define guidelines, standards, and interfaces for integrating and automating systems management applications. The *end-use dimension* defines a common user interface that is based on a programmable workstation. The *application dimension* provides a comprehensive approach to integrate the different applications and the systems management tasks needed to manage the applications. The management tasks are divided into 6 disciplines: business, change, configuration, operations, performance, and problem management. These disciplines are grouped into three levels according to their function: administration, coordination, and execution level. The *data dimension* defines an object-oriented database where information is shared between the different applications.

With the announcement of SystemView, IBM reinforced that Information/Management was a strategic product and made a statement of direction that Information/Management will become SystemView compliant.

With the announcement of Information/Management Version 5, IBM took the first step to establish the performance base that will enable the Information/Management product to support the SystemView system management strategy.

Information/Management Version 6 Release 1, with its utilization of ISPF Version 4 GUI is a significant step to SystemView end-use dimension compliance. For the first time, Information/Management provides a client/server interface.

Information/Management is enhanced to support remote access by customer and vendor client applications exploiting the new Information/Management client APIs on OS/2 and CICS/ESA platforms. The Information/Management CUA lookalike interface provides context-specific, user-modifiable action bars with modifiable pull-down menus and keylists for all Information/Management panels.

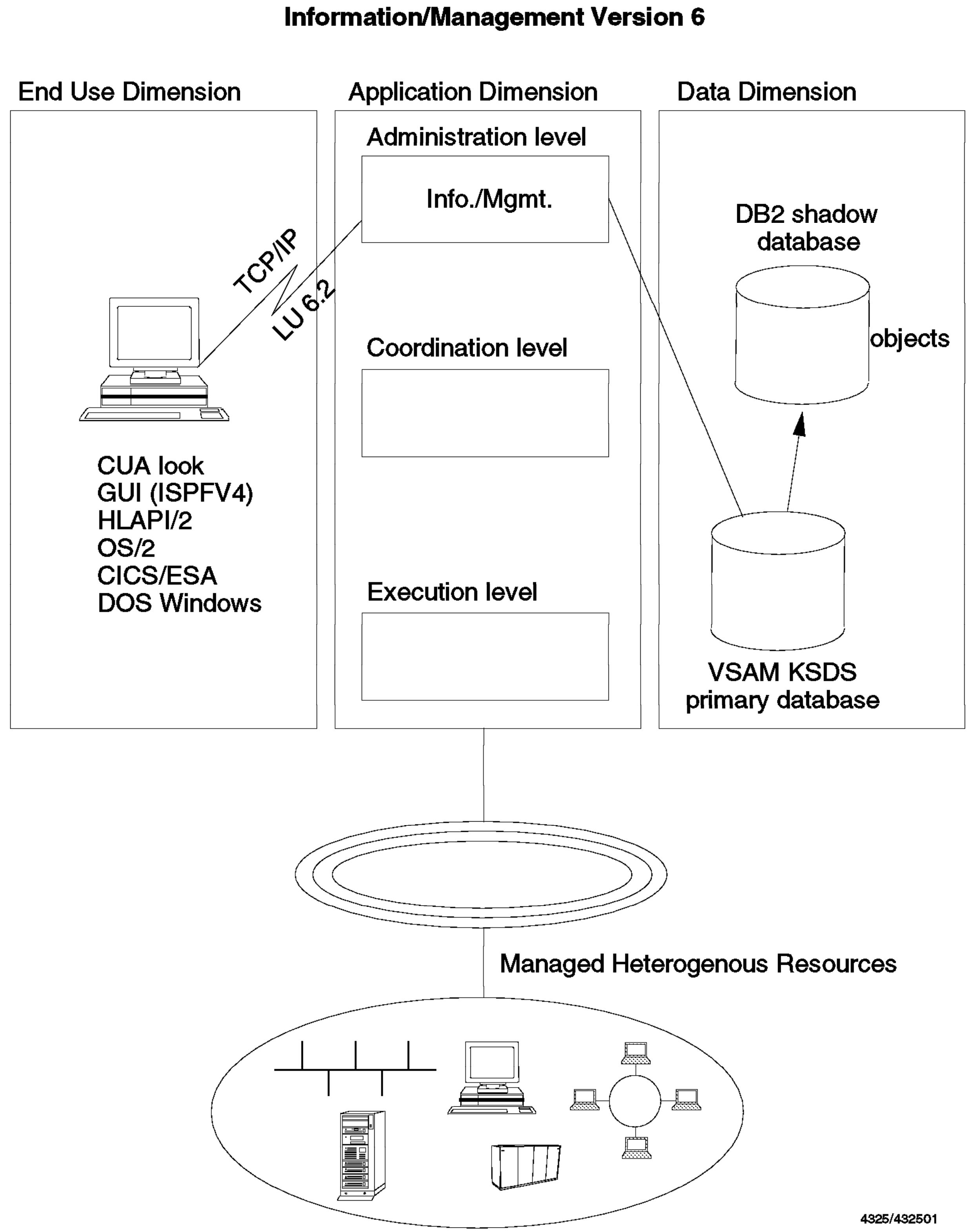


*Figure 3. Information/Management Version 6 Release 1*

Customers also have the ability to store Information/Management data in DB2 tables, for query and reporting, by using a new load utility. SQL access to the

Information/Management database is enabled through an interface provided by the IBM Information Warehouse* Framework.

# Chapter 2. Pre-Installation Tasks

This chapter provides information that will be useful in the planning stage, so that you know whether or not you have all the hardware and software that is required before proceeding with installing Information/Management Version 6 Release 1 or any of its new features. The product is shipped on up to four tapes as follows:

1. OY6101 - Base product
2. OY6105 - HLAPI/2 feature
3. OY6107 - US English feature
4. OY6108 - CICS* feature

## 2.1 Hardware Requirements

The base product requires a host processor that accommodates the software described in 2.2, “Software Requirements” on page 10. Information/Management has no dependencies on any new or changed hardware equipment.

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

To display color graphics reports, you will require a color graphics display that the Graphical Data Display Manager (GDDM*) supports. An IBM 3287 printer of equivalent will be required to print these reports.

Each HLAPI/2 client requires the following:

1. An IBM PS/2* or equivalent system capable of running the IB Operating System/2 (OS/2) Version 2.0 or higher, and Communications Manager/2 Version 1.0 or higher.
2. A fixed disk with 2MB free space to support:
    a. Installation and Maintenance Utility
    b. Run time parts
    c. Toolkit parts
    d. User′s guide and reference parts
3. Token Ring Adapter Card and network or a communication option to support LU 6.2 communication.
4. A minimum of 6MB of memory to operate HLAPI/2 and its corequisite software. However, we recommend that you have 12MB.

If you wish to implement Multisystem Database Access to share data across MVS/ESA systems, you will need to set up a global resource serialization (GRS) complex, which will require one of the following:

1. An IBM 3088 Multisystem Channel Communications Unit (MCCU)
2. Enterprise Systems Connection Architecture (ESCON*) channels operating in channel to channel (CTC) mode.

## 2.2 Software Requirements

The following software are prerequisites for the base product:

1. Multiple Virtual Storage/Enterprise Systems Architecture (MVS/ESA) System Product Version 4.2 or higher.
2. Multiple Virtual Storage/Data Facility Product (MVS/DFP*) Version 3.2 or higher.
3. System Modification Program/Extended (SMP/E*) Version 1.7 or higher
4. Interactive System Productivity Facility (ISPF) Version 3.5 or higher.

   Please note Version 4.1 is required if you wish to use the Graphical User Interface (GUI), or connect your workstation via TCP/IP.
5. Interactive System Productivity Facility/Program Development Facility (ISPF/PDF) Version 3.5 or higher (required only when using ISPF 3.5, ISPF 4.1 is a merge of the previous ISPF and PDF).

The optional features have software requirements summarized below:

1. The graphics function of the Report Format Facility
   a. Graphical Data Display Manager (GDDM) Version 2.2 or higher
   b. GDDM Presentation Graphics Feature (GDDM-PGF), Version 2.1.1 or higher
   c. OS PL/1 Library Version 2.3 or higher
2. The NetView* Bridge Adapter
   a. NetView Version 2.3
   b. OS PL/1 Library Version 2.3 or higher
3. Integration Facility interfaces
   a. NetView Version 2.3 for MVS/ESA
   b. Operations Planning and Control/Advanced - Event Manager Subsystem, Version 1.2
   c. OPC/A* - Production Control System Version 1.2
   d. OPC/A - Network Event Communicator Version 1.2
   e. Service Level Reporter (SLR) Version 3.2 or a subsequent release
   f. IBM SystemView Enterprise Performance Data Manager/MVS (EPDM) Version 1
4. Displaying DBCS characters
   a. Virtual Telecommunications Access Method (VTAM*) Version 3.3 with OY35982 or higher
   b. Each workstation must have its LOGMODE configured as follows:
      1) FORMAT=NEW
      2) The query bit of PSERVIC must be set
      3) The LANG parameter must be set to X′80′
5. HLAPI/2 Client workstation
   a. OS/2 Version 2.0 with ServicePak XR06055 or higher
   b. CM/2 Version 1.0
   c. A C language compiler that supports the 32-bit linkage convention
6. Remote Environment Server
   a. VTAM Version 3.3 with APAR OY35982 or higher
   b. IBM 3270/PC File Transfer Program for MVS/TSO

7. HLAPI CICS Interface

   a. CICS/ESA Version 3.3 or higher
   b. VTAM Version 3.3 with APAR OY35982 or higher

   The sample CICS application requires COBOL II Version 1.4 to run.

8. Multisystem Database Access

   a. VTAM Version 3.3 with APAR OY35982 or higher

9. DB2 Extract Facility

   a. IBM DATABASE 2*(DB2) Version 2.3 or higher

## 2.3 Space Requirements

The tables below provides details for the amount of DASD space required for installing the base product.

### 2.3.1 Base Product Target Libraries

| Data Set Name | DSORG | RECFM | LRECL | BLK SIZE | No. of BLKS | No. of DIR BLKS |
|---|---|---|---|---|---|---|
| BLM.V6R1M0.SBLMDICT | PO | FB | 80 | 6160 | 235 | 5 |
| BLM.V6R1M0.SBLMEXEC | PO | FB | 80 | 6160 | 230 | 20 |
| BLM.V6R1M0.SBLMFMT | PO | FB | 80 | 6160 | 850 | 20 |
| BLM.V6R1M0.SBLMMACS | PO | FB | 80 | 6160 | 140 | 5 |
| BLM.V6R1M0.SBLMMOD1 | PO | U | 0 | 6144 | 1000 | 50 |
| BLM.V6R1M0.SBLMPNLS | PO | FB | 80 | 6160 | 3000 | 400 |
| BLM.V6R1M0.SBLMSAMP | PO | FB | 80 | 6160 | 400 | 20 |
| BLM.V6R1M0.SBLMINST | PO | FB | 80 | 6160 | 230 | 25 |
| BLM.V6R1M0.SBLMSRC1 | PO | FB | 80 | 6160 | 125 | 5 |
| BLM.V6R1M0.SBLMTXT1 | PO | U | 0 | 6144 | 200 | 20 |
| BLM.V6R1M0.SBLM2API | PO | FB | 80 | 6160 | 150 | 10 |
| BLM.V6R1M0.SBLM2ENU | PO | FB | 80 | 6160 | 270 | 10 |

### 2.3.2 Base Product Distribution Libraries

| Data Set Name | DSORG | RECFM | LRECL | BLK SIZE | No. of BLKS | No. of DIR BLKS |
|---|---|---|---|---|---|---|
| BLM.V6R1M0.ABLMDICT | PO | FB | 80 | 6160 | 235 | 5 |
| BLM.V6R1M0.ABLMEXEC | PO | FB | 80 | 6160 | 230 | 20 |
| BLM.V6R1M0.ABLMFMT | PO | FB | 80 | 6160 | 850 | 20 |
| BLM.V6R1M0.ABLMMACS | PO | FB | 80 | 6160 | 140 | 5 |
| BLM.V6R1M0.ABLMMOD1 | PO | U | 0 | 6144 | 1650 | 250 |
| BLM.V6R1M0.ABLMPNLS | PO | FB | 80 | 6160 | 3000 | 400 |
| BLM.V6R1M0.ABLMSAMP | PO | FB | 80 | 6160 | 400 | 20 |
| BLM.V6R1M0.ABLMINST | PO | FB | 80 | 6160 | 230 | 25 |
| BLM.V6R1M0.ABLMSRC1 | PO | FB | 80 | 6160 | 125 | 5 |
| BLM.V6R1M0.ABLMTXT1 | PO | U | 0 | 6144 | 200 | 20 |
| BLM.V6R1M0.ABLM2API | PO | FB | 80 | 6160 | 150 | 10 |
| BLM.V6R1M0.ABLM2ENU | PO | FB | 80 | 6160 | 270 | 10 |

The tables below provides details for the amount of DASD space required for installing HLAPI CICS.

### 2.3.3 HLAPI CICS Target Libraries

| Data Set Name | DSORG | RECFM | LRECL | BLK SIZE | No. of BLKS | No. of DIR BLKS |
|---|---|---|---|---|---|---|
| BLM.V6R1M0.SBLMMOD1 | PO | U | 0 | 6144 | 100 | 10 |
| BLM.V6R1M0.SBLMTXT1 | PO | U | 0 | 6144 | 200 | 10 |
| BLM.V6R1M0.SBLMSAMP | PO | FB | 80 | 6160 | 50 | 10 |
| BLM.V6R1M0.SBLMSRC1 | PO | FB | 80 | 6160 | 50 | 5 |

### 2.3.4 HLAPI CICS Distribution Libraries

| Data Set Name | DSORG | RECFM | LRECL | BLK SIZE | No. of BLKS | No. of DIR BLKS |
|---|---|---|---|---|---|---|
| BLM.V6R1M0.ABLMMOD1 | PO | U | 0 | 6144 | 100 | 10 |
| BLM.V6R1M0.ABLMTXT1 | PO | U | 0 | 6144 | 200 | 10 |
| BLM.V6R1M0.ABLMSAMP | PO | FB | 80 | 6160 | 50 | 10 |
| BLM.V6R1M0.ABLMSRC1 | PO | FB | 80 | 6160 | 50 | 5 |

### 2.3.5 Preventive Maintenance Checks

Before proceeding with the installation, as documented in the *Program Directory for Information/Management*, please obtain a copy of the PSP bucket, which provides information that has become known after the publication of the Program Directory.

The PSP Upgrade bucket is INFO610. There are four subset IDs, as follows:

1. HOY6101 - Base product
2. HOY6107 - US English Feature
3. JOY6105 - HLAPI/2 Feature
4. JOY6108 - CICS Feature

# Chapter 3. Installation

This chapter is structured as follows:

1. Checklist for installing a prime Version 6 Release 1 system. This checklist will get you up and running very fast if you′re already experienced with Information/Management.

   **Note:** After performing these steps you′ll have a test Information/Management system with an empty database. Follow the migration instructions to make data from an existing database available (refer to Chapter 4, "Migration from Previous Versions" on page 19).

2. Installation with Installation Tailoring Facility (ITF) You can use ITF to build your installation JCL. ITF will take care of most of the JCL complexities for you.

   It will also build your session parameters member and BLX-SP startup procedure. Refer to 3.2, "Installation Tailoring Facility (ITF)" on page 16.

## 3.1 Prime System Installation

Please follow these steps to install an operational Information/Management Version 6 Release 1 system with an empty database. You can test your newly installed system and switch to production later (refer to Chapter 4, "Migration from Previous Versions" on page 19).

### 3.1.1 Installation Tasks List

Read the entire checklist. An IPL will be required and it must be planned ahead of time. If you′re running Information/Management Version 5 Release 1, do not attempt to avoid an IPL by re-using Version 5 Release 1 APF library names and the BLX0 subsystem name. It will fail and force you to perform an IPL to restart your Information/Management system. We do recommend that you use the BLX1 subsystem name (SSN) and a new set of APF libraries. This will allow for parallel execution of Information/Management Version 5 Release 1 and Version 6 Release 1. If you′ve automated Information/Management Version 5 Release 1 console operations you must update your automation procedures with the new SSN (BLX1). Use the test period to verify proper operation and perform any required automation changes.

1. Define your SDDS, SDIDS and SDLDS data sets.

   Copy job BLGDATAB from SBLMSAMP to your JCL library. Modify the JCL statements to suit your installation requirements. Code the proper data set names, volume serials and space information. Submit the job.

2. Define your DICTDS (dictionary data set).

   Copy job BLGDICT from SBLMSAMP to your JCL library. Modify the JCL statements to suit your installation requirements. Code the proper data set names, volume serials and space information. Submit the job.

3. Define your read panel data set.

   Copy job BLGRPNL from SBLMSAMP to your JCL library. Modify the JCL statements to suit your installation requirements. Code the proper data set names, volume serials and space information. Submit the job.

4. Define your write panel data set.

   Copy job BLMWPNL from SBLMSAMP to your JCL library. Modify the JCL statements to suit your installation requirements. Code the proper data set names, volume serials and space information. Submit the job. This data set will be used for panel tailoring.

5. Determine VSAM buffer pool sizes.

   Use LISTCAT to get the CISIZE values for each of the data and index components of the VSAM clusters. (This is required due to the fact that VSAM may use a different CI value than the one you specified on the DEFINE CLUSTER when defining the data sets). You can run LISTCAT either from the TSO command prompt or with a batch job. The easier way to obtain all the information is to use the following AMS command:

   ```
   LISTCAT LEVEL('BLM.V6R1M0') ALL
   ```

   Substitute BLM.V6R1M0 with the higher level qualifiers for your Information/Management Version 6 Release 1 VSAM data sets.

6. Update VSAM definitions member BLXVDEF.

   Job BLXVDEF is used to create Information/Management VSAM definitions. It uses the BLDVRP macro to build VSAM LSR resource pools. These resource pools contain buffer pools. You may have several buffer pools in each resource pool. You must have buffer pools in your resource pool to fit the CI size of each VSAM cluster component (Data and Index).

   - Copy job BLXVDEF from SBLMSAMP to your JCL library.
   - Modify the JCL statements to suit your installation requirements.
   - Update all data set names to match your Information/Management system configuration.
   - Code each LSR buffer pool size matching the CI size of the component assigned to that buffer pool.

7. Copy the BLX1 procedure to a system library.

   Copy the BLX1PROC member from SBLSAMP to your SYS1.PROCLIB (or to your installation defined procedure library). Modify the STEPLIB and the BLXPARMS DD statements DSNAME parameters to match your load library and parameter data set names.

8. Make BLX1 a subsystem name.

   This is achieved by updating member IEFSSNxx of SYS1.PARMLIB, where xx is an installation-dependent suffix.

   If you want to run a production Information/Management Version 5 Release 1 system with a test Version 6 Release 1 system, then you must have entries for both BLX0 and BLX1 in this member.

9. Make your SBLMMOD1 library APF authorized.

   This is achieved by updating member IEAAPFxx of SYS1.PARMLIB. Again xx is an installation suffix.

   If you want to run a production Information/Management &v5. system alongside a test Version 6 Release 1 system, then you must have two sets of libraries defined in this member.

10. Make BLXSSP00 non-swappable.

This is achieved by updating member SCHEDxx of SYS1.PARMLIB. Again xx is an installation suffix.

11. Perform a system IPL.

    An IPL of your system will be required at this point in the installation process.

12. Start the service provider address space (BLX1)

    Issue the MVS operator command `S BLX1` from the operator console (or from your TSO session).

    Verify that the BLX-SP starts correctly and stays active by looking at the SYSLOG data set.

13. Create your session parameters member.

    This will assemble and link-edit a session parameters member, BLGSESxx. This member is used to tell Information/Management the names of the VSAM data sets for your installation requirements. You can choose a two digit suffix, xx, for the load module created or accept the default of **00**.

    Copy job BLGALSPM from SBLMSAMP to your JCL library. Change the JCL statements to suit your installation requirements. Code the proper data set names. Copy the session parameter member BLGSES00 from SBLMSAMP to your JCL library (or other suitable library). Modify the 00 suffix to suit your installation requirements.

14. Initialize your SDDS (structured description data set).

    Copy job BLGUT2J from SBLMSAMP to your JCL library. (Rename to something like UT2SDDS  when copying). Modify the JCL statements to suit your installation requirements. Change the BLGSD DD statement DSNAME parameter to match your SDDS data set name. Submit the job.

15. Initialize your SDIDS (structured description index data set).

    Copy job BLGUT2J from SBLMSAMP to your JCL library. (Rename to something like UT2SDIDS when copying). Modify the JCL statements to suit your installation requirements. Change the BLGSD DD statement DSNAME parameter to match your SDIDS data set name. Submit the job.

    **Note:** BLGUT1 should be used to build a valid SDIDS from SDDS records. This is required when you already have an active database. With an empty SDDS, creating the SDIDS control record is the only requirement, hence BLGUT2 could be used.

16. Initialize your SDLDS (structured description log data set)

    This is your recovery log data set. You′ll need one for production work. However, a SDLDS is not mandatory for a test system, and you may choose to not have one (in which case, the reference to this data set will need to be removed from the session parameter member, BLGSESxx).

    Copy job BLGUTRJ from SBLMSAMP to your JCL library. Modify the JCL statements to suit your installation requirements. Change the BLGSL DD statement DSNAME parameter to match your SDLDS data set. Submit the job.

17. Initialize your WPANLDS (write panel data set).

    Copy job BLGUT2J from SBLMSAMP to your JCL library. (Rename to something like UT2WPAN  when copying). Modify the JCL statements to suit your installation requirements. Change the BLGSD DD statement DSNAME parameter to match your WPANELS data set name. Submit the job.

18. Load your RPANLDS (read panels data set).

    Copy job BLGUT6J from SBLMSAMP to your JCL library. Modify the JCL statements to suit your installation requirements. Change the BLGPDS and the BLGPNLS DD statements to match your input panel PDS (SBLMPNLS) and your output panel VSAM data set (IBMPNLS). Submit the job.

19. Initialize your DICTDS (dictionary data set).

    Copy job BLGUT5J from SBLMSAMP to your JCL library. Modify the JCL statements to suit your installation requirements. Change the BLGSWDS and the BLGDICT DD statements to match your input s-words PDS (SBLMDICT) and your output dictionary VSAM data set (DICT).

20. Create an ISPF option for Information/Management.

    Information/Management should now be available for use, but you will need to set up an option on your ISPF primary selection panel to invoke it. The line below gives you an example. Please note that the value in the brackets for the SESS parameter needs to match the suffix you chose when you ran job BLGALSPM.

    ```
    I,′PGM(BLGINIT) PARM(SESS(xx)) NEWAPPL(BLGO)′
    ```

You′ve now completed the basic installation process.

## 3.2 Installation Tailoring Facility (ITF)

The Installation Tailoring Facility (ITF) of Information/Management Version 6 Release 1 is designed to help users tailor Information/Management. The tailoring is done through interactive ISPF panels that prompt the user for the necessary control information.

The Installation Tailoring Facility (ITF) can be used to create or modify the JCL or data to do the following on the base Information/Management setup:

1. Define the required Information/Management VSAM data sets, for example, SDDS, SDIDS, SDLDS, DICTDS, RPANLDS and WPANLDS
2. Define session-parameter members
3. Specify BLX-SP parameters members
4. Specify BLX-SP startup procedures
5. Specify BLX-SP VSAM resource definition members
6. Link-edit the graphics and the NetView Bridge Adapter
7. Implement features for national language support

### 3.2.1 Using Installation Tailoring Facility (ITF)

The Installation Tailoring Facility (ITF) feature must be customized to work in the ISPF environment. The following needs to be done:

1. Include the SBLMEXEC data set in your SYSPROC concatenation, ensuring that the SBLMEXEC data set is compatible with other data sets in the concatenation. The SBLMEXEC data set is pre-allocated as partitioned, with DCB parameters RECFM=FB and LRECL=80 (fixed blocked record format and 80

bytes record length). Please refer to 2.3.1, “Base Product Target Libraries” on page 11 for additional information.

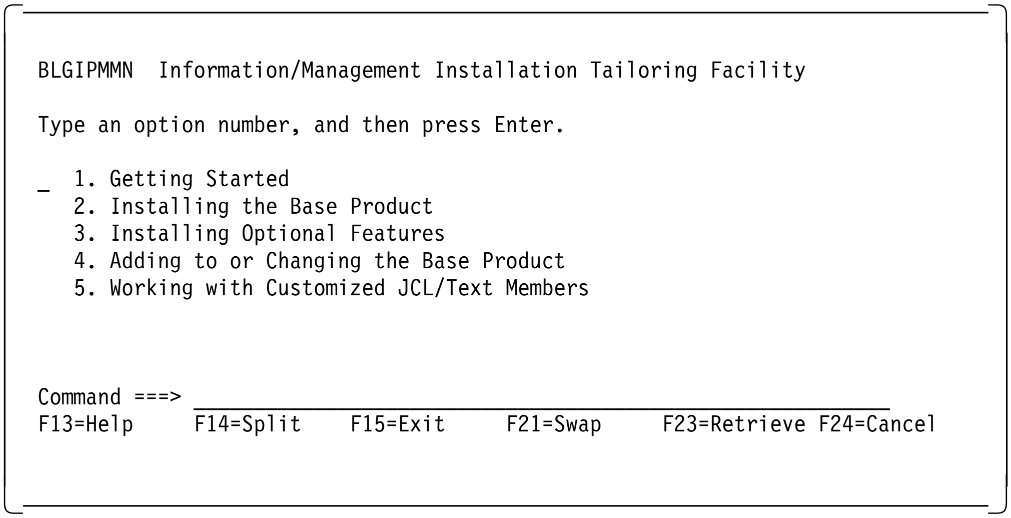
2. Invoke the Installation Tailoring Facility (ITF) by doing one of the following:

   - Issuing the command `BLGISTRT hlqualf` on the TSO Ready Screen.
   - Using the option `xx,CMD(BLGISTRT hlqualf)`, in the ISPF main panel. This option allows the user to drive the application in split screen mode, with another ISPF application, such as SDSF, on the other screen.

   Where `hlqualf` is the high-level qualifier of the SBLMINST data set assigned during the initial SMP/E installation. The default is `BLM.V6R1M0`. The Installation Tailoring Facility (ITF) uses `hlqualf` as a prefix for all Information/Management data sets, such as the SYSLIB and SYSLMOD, that it uses during JCL processing. If different `hlqualf` values are used for Information/Management data sets, then the JCL generated by the Installation Tailoring Facility (ITF) must be reviewed and manually changed before execution. The JCL is stored in data set `SBLMJCL1` by the Installation Tailoring Facility (ITF).

   **Note:** Split screen mode will hide some lines when messages, such as error messages, are displayed.

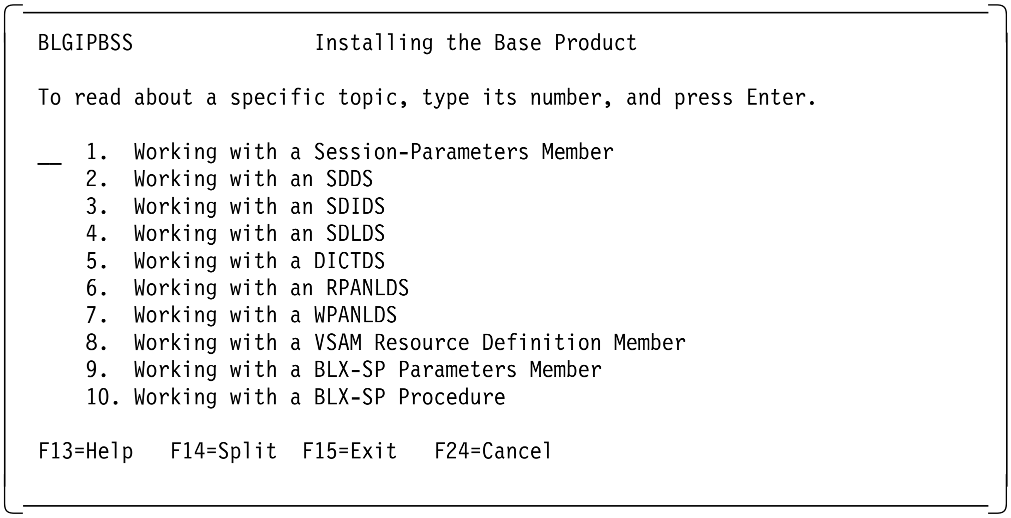
The following screen is presented after the Installation Tailoring Facility (ITF) is invoked. `Option 1` provides an extensive online help facility. The user can use this option as an online tutorial for the Installation Tailoring Facility (ITF).



```
BLGIPMMN  Information/Management Installation Tailoring Facility

Type an option number, and then press Enter.

_   1. Getting Started
    2. Installing the Base Product
    3. Installing Optional Features
    4. Adding to or Changing the Base Product
    5. Working with Customized JCL/Text Members



Command ===> ___________________________________________________________
F13=Help     F14=Split    F15=Exit     F21=Swap     F23=Retrieve F24=Cancel
```

*Figure 4. Panel BLGIPMMN - Installation Tailoring Facility (ITF) Main Panel*

The following figure illustrates some of the options available when setting up the Information/Management database. The screen is part of the online help facility.



```
BLGIPBSS              Installing the Base Product

To read about a specific topic, type its number, and press Enter.

__  1.  Working with a Session-Parameters Member
    2.  Working with an SDDS
    3.  Working with an SDIDS
    4.  Working with an SDLDS
    5.  Working with a DICTDS
    6.  Working with an RPANLDS
    7.  Working with a WPANLDS
    8.  Working with a VSAM Resource Definition Member
    9.  Working with a BLX-SP Parameters Member
    10. Working with a BLX-SP Procedure

F13=Help   F14=Split  F15=Exit   F24=Cancel
```

*Figure 5. Panel BLGIPBSS - Installing the Base Product Help Panel*

The user still needs to perform the following tasks after allocating Information/Management data sets and setting up Information/Management parameters with the Installation Tailoring Facility (ITF). Refer to 3.1.1, “Installation Tasks List” on page 13 for a detailed explanation of each one.

1. Initialize the WPANLDS (write panels data set)
2. Initialize the RPANLDS (read panels data sets)
3. Initialize the SDLDS (structured description log data set)
4. Initialize the SDIDS (structured description index data set)

   **Note:** BLGUT1 should be used to build a valid SDIDS from SDDS records. This is required when you already have an active database. With an empty SDDS, creating the SDIDS control record is the only requirement, hence BLGUT2 could be used.

5. Copy the BLX1 procedure to a system library.
6. Perform all SYS1.PARMLIB updates

You can use the list from 3.1.1, “Installation Tasks List” on page 13 as a checklist to verify that you covered all required installation steps when using the Installation Tailoring Facility (ITF).

After you have finished with the Installation Tailoring Facility (ITF) process, you′ll have a library with all the Information/Management installation JCL tailored for your system. This JCL can be edited and submitted when necessary.

# Chapter 4. Migration from Previous Versions

This section covers specific migration steps from Version 4 and Version 5 Release 1 of Information/Management.

Please refer to 3.1.1, "Installation Tasks List" on page 13 for a complete installation tasks list. Use it as a verification checklist for the migration process.

Because migration from Version 4 is more complex than from Version 5 Release 1, the two sections will be discussed separately.

Items discussed here are required in order to migrate successfully to Version 6 Release 1. For detailed information about the new functions or enhancements of Version 6 Release 1, please refer to Chapter 5, "Installing the Client/Server Interfaces" on page 25.

## 4.1 Initial Migration Steps

The following activities are necessary for migration from both Version 4 and Version 5 Release 1. The Installation Tailoring Facility (ITF) of Information/Management Version 6 Release 1 can be used to generate the JCL that performs these tasks (this is specially useful for users who won′t use default parameters, since the Installation Tailoring Facility (ITF) will simplify the process):

- Define the Information/Management data sets with recommended VSAM attributes.
- Create the session-parameter member member.
- Set up started task parameters member (BLX100).
- Update VSAM definitions member BLXVDEF.

The data sets defined above must be initialized with the corresponding Information/Management utilities. Refer to 3.1.1, "Installation Tasks List" on page 13.

The IBM-supplied BLX-SP procedure is shown in Figure 6 on page 20.

- BLX1 - is the procedure name. Keyword PRM=00 indicates that BLXPRM (see below) member BLX100 contains parameters for the started task.
- BLXSSP00 - identifies the Information/Management program used by the procedure. This module must reside in an APF library.
- STEPLIB - the Information/Management module library. This DDNAME is not required if the LINK LIST concatenation is used to access the Information/Management modules. The user Information/Management session parameters - BLGSESxx - can also be placed in this library.

  NOTE: We recommend that you use STEPLIB instead of the system link list and that the session parameters member BLGSESxx be placed in the BLXPRM library.
- BLXPRM - the BLX-SP parameter library. This is where the BLX100 member resides. Usually this DD points to the SBLMSAMP data set.

```
//BLX1     PROC PRM=00
//BLXSPCAS EXEC PGM=BLXSSP00,REGION=6M,TIME=1440,PARM=&PRM
//STEPLIB  DD   DSN=BLM.V6R1M0.SBLMMOD1       APF AUTHORISED
//BLXPRM   DD   DISP=SHR,BLM.V6R1M0.SBLMSAMP  STC BLX-SP PARMS
```

*Figure 6. Procedure to Start Information/Management Version 6 Release 1*

## 4.2 Migrating from Version 4

Information/Management Version 6 Release 1 differs in many aspects from Version 4, the most important being the BLX-SP service provider address space. This address space is an MVS subsystem. Defining this subsystem to MVS requires many changes (you must involve the MVS support person for this task). Due to the above, the migration from Version 4 to Version 6 Release 1 requires some extra effort.

There are also additional considerations for Version 4 Release 2 users. The migration process involves changes to the following components:

1. SYS1.PARMLIB
2. SYS1.PROCLIB
3. BLGSESxx Parameters
4. BLXVDEF - VSAM definition pools
5. BLX100 - BLX-SP parameters
6. RACF* definitions

Additional considerations apply if Information/Management Version 4 and Version 6 Release 1 are installed in different MVS systems. These are included in Table 1.

*Table 1 (Page 1 of 2). Version 4 to Version 6.1 Migration Checklist*

| Item | Action | Customization process |
|---|---|---|
| BLX1 subsystem | Make the BLX-SP known to MVS | You must define a new subsystem to MVS by changing the following in SYS1.PARMLIB.<br>• Add the new subsystem name BLX1 to IEFSSNxx.<br>• Add the BLX-SP program name (BLXSSP00) to the Program Properties Table (SCHEDxx) marking it non-swappable<br>• APF authorize the Version 6 Release 1 module library (add its data set name to IEAAPFxx). |
| RACF | Make RACF aware of Information/Management | Define the GINFOMAN class to RACF |
| PROCLIB | Set up the BLX-SP JCL procedure | Copy JCL for BLX-SP to SYS1.PROCLIB. Ensure Module library in the STEPLIB is APF authorized |
| BLX1 started task parameters | Set up BLX100 member | Set up parameters as shown in the SBLMSAMP library, member BLX100. |

*Table 1 (Page 2 of 2). Version 4 to Version 6.1 Migration Checklist*

| Item | Action | Customization process |
|---|---|---|
| VSAM resource pool definitions | Set up buffer pools for VSAM data sets. | VSAM parameters must be set up in job BLXVDEF, from the SBLMSAMP library. This job assembles and link-edits the VSAM resource pool definitions to a Version 6 Release 1 load module library. Use the procedure explained in 3.1.1, "Installation Tasks List" on page 13 to find the proper values for the buffer sizes. |
| BLGSESxx session parameters | Remove ENQLVL, LSR, BREPORT, IREPORT and PRODUCT parameters | Ensure that the session parameters are compatible with and use the functions of Version 6 Release 1. The LSR definitions have been moved to BLXVDEF, which defines VSAM resource pools. |
| List processor exit - BLG01396 | Review processing changes | User TSPs may require some changes. Review manual *Panel Modification Facility Guide and Reference*, SC34-4457. |
| Macros | Add PDF editor support | Add BLGISMAC to the SYSPROC concatenation to provide access to the Information/Management edit macros |
| Profile variables | Support user defined profile variables | BLGPVARS must be added to the ISPPLIB concatenation. |
| NPDA Interface | BLMVATSR and BTNCNMBD modified | Customize the new macros according to the installation′s requirements. See ***Making Major Changes to the Interface*** in the Appendix section of *Planning and Installation Guide.* |
| Table command | Support the new minimum truncation for the TABLE command (TA) | Modify TSPs and SRCs that use the old minimum truncation of T to use TA. |
| Panel changes | Modified and deleted panels | PMF should be used to migrate panels created under Version 4 to Version 6 Release 1 format. Review the *Planning and Installation Guide* in the Appendix on new, changed and removed panels. |
| VSAM allocation parameters | Change allocation parameters | Allocate the VSAM data sets with the new recommended attributes and use the REPRO function to copy the old data sets to the new ones. There are performance benefits in using the new attributes for data set allocation. |

Items shown in Table 1 on page 20 apply to all users migrating from Version 4. The following are additional considerations for customers migrating from Version 4.1.

- Updated compression algorithm - If the file-time processing panels have been modified, the last dialog end collected for each record type must also have an S-word index of 0CF1.

The following are additional considerations for customers migrating from Version 4 Release 2.

- HLAPI considerations.

  PIDTs modified for DBCS support - BLGUT8 should be used to migrate PIDTs created by the installation.

  If the HLAPI and LLAPI are used, your application must be link-edited with AMODE(31) and RMODE(ANY).

If there are user applications using either the HLAPI or the LLAPI and they are link-edited with AMODE(24) and RMODE(24),a link must be done to the HLAPI or LLAPI program, and not a LOAD or a CALL

- LLAPI considerations.

  Expanded freeform searches - if there are applications that performs freeform searches using individual freeform arguments that are greater than 16 characters, modify the declaration in program interface argument table (PIAT) in the application, or the application can be link-edited to include the modified PIAT.

## 4.3 Migrating from Version 5 Release 1

Information/Management Version 6 Release 1 has been designed to make the migration from Version 5 Release 1 a relatively simple task. Migration to implement the Version 5 Release 1 system in a Version 6 Release 1 can be done with minimal changes, if the new Version 6 Release 1 functions will not be used immediately.

The detailed information on the planning, installation and customization of Information/Management Version 6 Release 1, is discussed in 3.1, “Prime System Installation” on page 13. This section will cover the steps of integrating the Information/Management Version 5 Release 1 database with Version 6 Release 1. Refer to Table 2 before attempting to use the Version 5 Release 1 database and data sets with Version 6 Release 1.

*Table 2. Version 5.1 to Version 6.1 Migration Checklist*

| Item | Action | Customization Process |
|---|---|---|
| BLX-SP | Change from BLX0 to BLX1 | Define the new subsystem. Modify all documented procedures to refer to the new name. |
| Table command | Support the new minimum truncation for the TABLE command (TA) | Modify TSPs and SRCs that use the old minimum truncation of T to use TA. |
| HLAPI | Convert PIDTs for DBCS support | BLGUT8 should be used to convert installation-defined PIDTs |
| Panel changes | Modified and deleted panels | PMF should be used to migrate panels created under Version 5 Release 1 to Version 6 Release 1. Review the *Planning and Installation Guide* in the Appendix on new, changed and removed panels. |
| VSAM allocation parameters | Change allocation parameters | Allocate the VSAM data sets with the new recommended attributes and use the REPRO function to copy the old data sets to the new ones. There are performance benefits in using the new attributes for data set allocation. |
| Install IBM supplied Panels | Run BLGUT5 | Run this supplied utility to ensure that your IBM Read panels data set has the latest panels. |
| Supplied Dictionary Entries | Run BLGUT6 | Run this supplied utility to ensure that your Dictionary data set has the latest base dictionary contents. |
| Automation | Change console automation | If you have console automation at your site that issues STARTS, STOPS or QUERIES for the started task BLX0, then these will need to be changed to BLX1 (assuming you use the default with Version 6 Release 1). |

As previously mentioned, the above discussion ensures a smooth migration from Information/Management Version 5 Release 1 to Version 6 Release 1. Refer to Chapter 5, "Installing the Client/Server Interfaces" on page 25 for implementing the new functions.

## 4.4 Starting Information/Management

The following are the steps to activate Information/Management Version 6 Release 1. They should be done after the activities listed either in Table 1 on page 20 or Table 2 on page 22.

1. Assemble the BLXVDEF member and link-edit it to Version 6 Release 1 module library (BLM.V6R1M0.SBLMMOD1).
2. Assemble the BLGSESxx member and link-edit it to the above-mentioned module library.
3. Terminate the BLX0 procedure for Version 5 Release 1 if it is active.
4. An IPL is required now to make the module libraries APF authorized and to activate the new BLX1 SSN.
5. Start the BLX1 procedure, which is your Version 6 Release 1 started task.
6. Change the logon procedure to point to the Version 6 Release 1 module library. If the Linklist is going to be used, ensure that the Information/Management Version 6 Release 1 module is accessed prior to Version 5 Release 1, then do an LLA refresh.
7. Log on to TSO using the updated logon procedure and you can then access and start using Version 6 Release 1.

# Chapter 5. Installing the Client/Server Interfaces

Two client/server interfaces are available with Information/Management Version 6 Release 1:

- ISPF Client/Server

  Also known as ISPF Graphical User Interface, or GUI for short. It provides a client ISPF task on a workstation running the OS/2 operating system or the DOS operating system and Microsoft Windows**.

  An active TSO user ID connected to a 3270 terminal or terminal emulator is required to initiate and maintain this client active.

- HLAPI/2

  This API provided by Information/Management Version 6 Release 1 allows access to Information/Management HLAPI functions (thus providing access to the Information/Management databases) from a workstation running the OS/2 operating system.

  A TSO user ID is not required for this client to operate.

Information for installing both client programs in the workstation are provided in *Jumping out of the Box: New Interfaces and Features of Information/Management Version 6 Release 1*, GG24-4325.

## 5.1 Remote Environment Server (RES) Module

The Information/Management Version 6 Release 1 *Program Directory for Information/Management* lists an optional installation step as described below. It is required for installing the HLAPI/2 interface and the CICS interface.

### 5.1.1 Link-editing the RES

This step is a prerequisite task for users wishing to implement either HLAPI/2 or the CICS HLAPI interface. The sample installation JCL library SBLMSAMP provides a job, BLMYSCSL, which performs a link-edit of the Remote Environment Server load module BLMYSCSC outside of SMP/E control. This integrates your APPC module ATBPBI as the first CSECT of the RES module.

## 5.2 Preparing APPC/MVS Definitions

### 5.2.1 Checklist

These steps apply to both the ISPF Client/Server and HLAPI/2 clients.

Establishing communication involves setup activities in two places: first in APPC/MVS* on the host machine, and second in CM/2 on the workstation. The diagram in Figure 7 on page 26 shows where parameters at the two ends of the link need to match in order to communicate.

The CM/2 definitions are described in *Jumping out of the Box: New Interfaces and Features of Information/Management Version 6 Release 1*, GG24-4325.

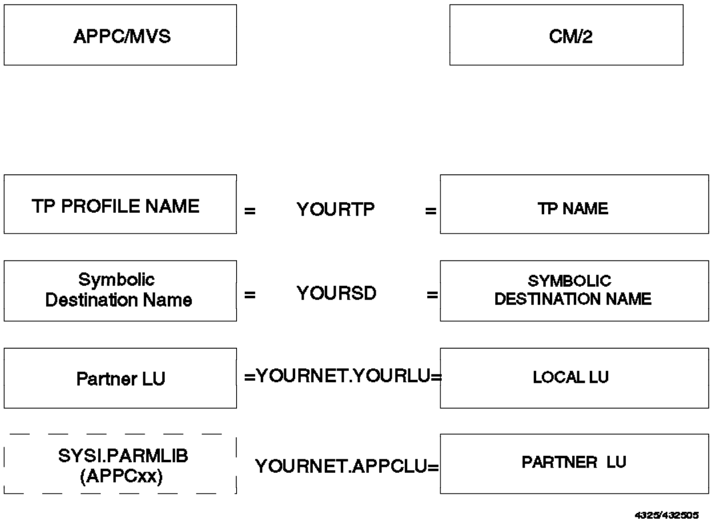


*Figure 7. APPC/MVS and CM/2 Cross Reference Summary*

The following lists the steps that are involved in preparing APPC/MVS. Read carefully the whole procedure before starting.

1. Verify the VTAM/NCP definitions for your workstation.

   The workstation that will connect to the host requires an LU defined to VTAM as LU 6.2 capable (rather than LU 2.1 capable for 3270 host sessions).

   As well as being defined in VTAM and NCP, this LU must be known to APPC/MVS. It must be listed in the parmlib member APPCxx. Refer to Figure 20 on page 34, line LUADD ACBNAME, for an example.

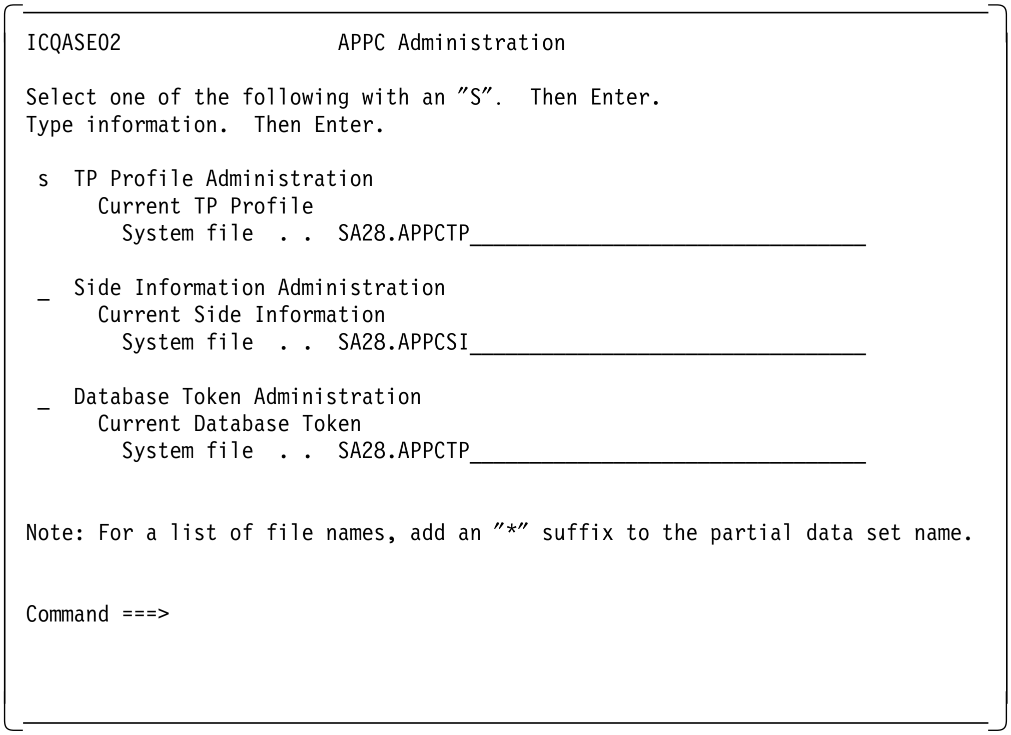
2. Invoke the APPC/MVS administration dialogs.

   There is likely to be an ISPF option on your system that calls this. If there is not, then assuming all the APPC data sets are available to your logon procedure, enter ICQASRM0 from ISPF option 6 to invoke the APPC/MVS administration dialogs.

   There are three selectable choices on this menu (see Figure 8 on page 27). We shall use two of these, one to define a TP profile and another to define side information for our needs.

3. Define a TPNAME in the APPCTP file in APPC/MVS.

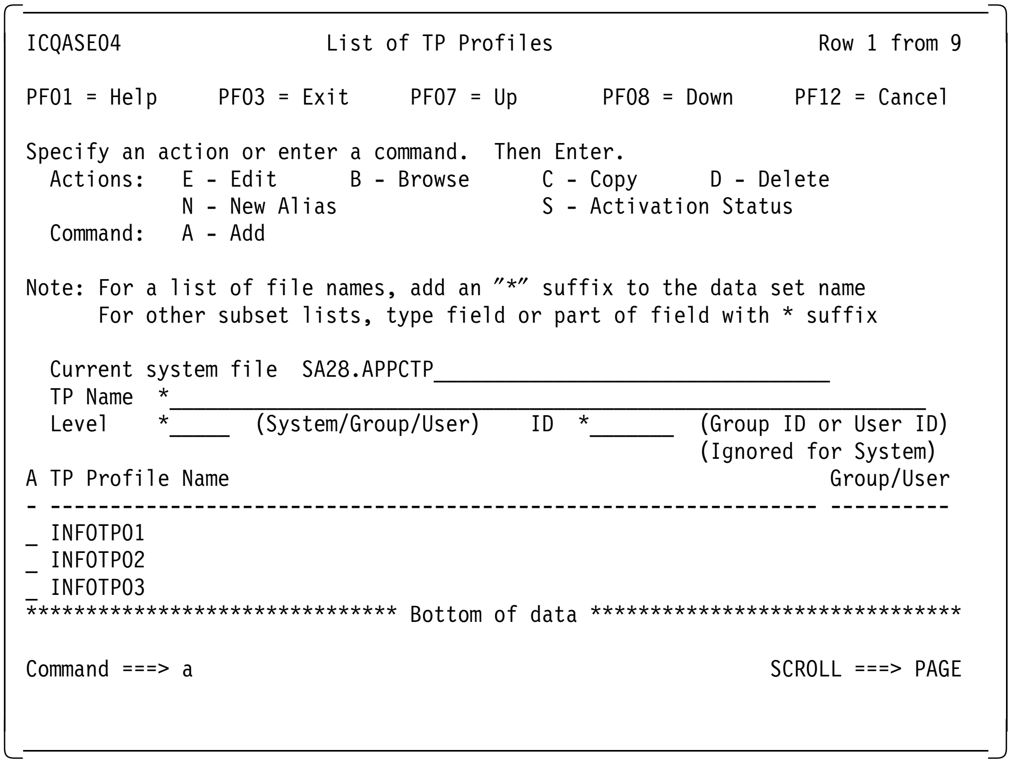
   a. Select TP Profile Administration on panel ICQASE02 in Figure 8 on page 27 and press the Enter key.



```
ICQASE02                  APPC Administration

Select one of the following with an ″S″.  Then Enter.
Type information.  Then Enter.

 s  TP Profile Administration
      Current TP Profile
        System file  . .  SA28.APPCTP_________________________________

 _  Side Information Administration
      Current Side Information
        System file  . .  SA28.APPCSI_________________________________

 _  Database Token Administration
      Current Database Token
        System file  . .  SA28.APPCTP_________________________________


Note: For a list of file names, add an ″*″ suffix to the partial data set name.


Command ===>
```

*Figure 8. Panel ICQASE02 - The APPC/MVS Administration Menu*

b. Enter A on the command line of panel ICQASE04 in Figure 9 to create a new TP profile and press the Enter key.



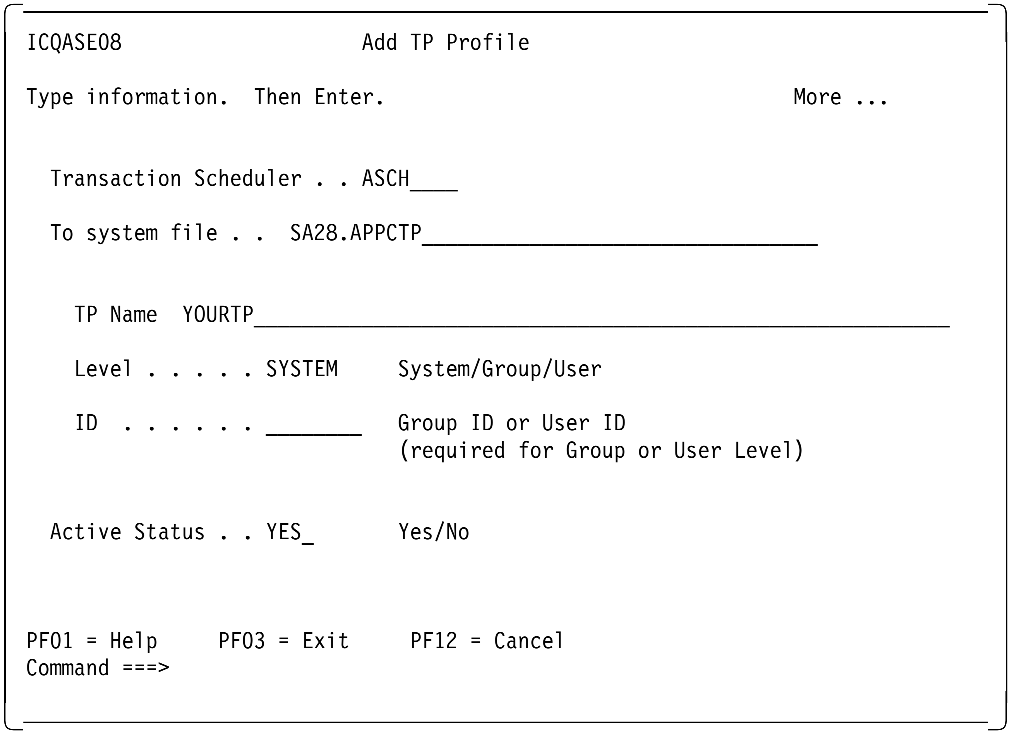
```
ICQASE04                 List of TP Profiles                       Row 1 from 9

PF01 = Help     PF03 = Exit     PF07 = Up       PF08 = Down     PF12 = Cancel

Specify an action or enter a command.  Then Enter.
  Actions:   E - Edit      B - Browse      C - Copy      D - Delete
             N - New Alias                 S - Activation Status
  Command:   A - Add

Note: For a list of file names, add an ″*″ suffix to the data set name
      For other subset lists, type field or part of field with * suffix

  Current system file  SA28.APPCTP_________________________________
  TP Name  *_______________________________________________________________
  Level    *_____  (System/Group/User)    ID  *_______  (Group ID or User ID)
                                                        (Ignored for System)
A TP Profile Name                                                  Group/User
- ---------------------------------------------------------------- ----------
_ INFOTP01
_ INFOTP02
_ INFOTP03
******************************* Bottom of data *******************************

Command ===> a                                                SCROLL ===> PAGE
```

*Figure 9. Panel ICQASE04 - The APPC/MVS TP Profiles Panel*

c. On panel ICQASE08 in Figure 10 on page 28, you will need to choose a TPNAME of your own to enter here. There are no restrictions other than length on what can be entered here, but the important point to note is that this matches the TPNAME you refer to in the workstation definitions later. These fields are case sensitive. You might want to use uppercase throughout as a naming convention. On the field labelled Level, type SYSTEM and press the Enter key.

Record your TPNAME below for future reference:

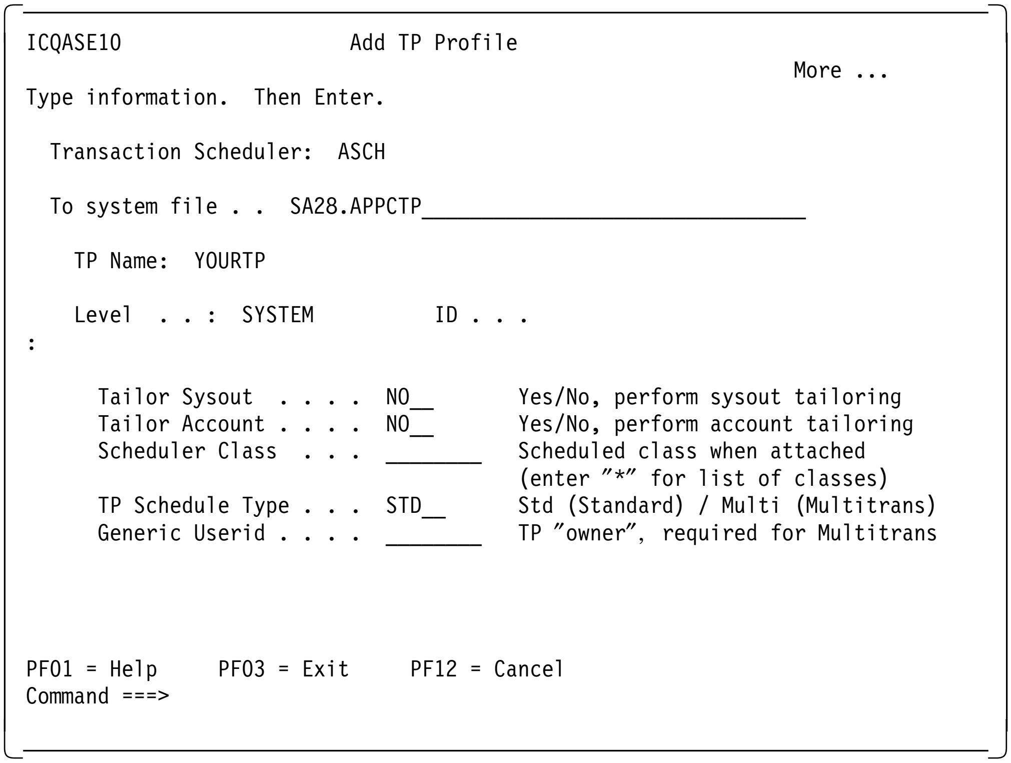
_______________________________________________



```
ICQASE08                    Add TP Profile

Type information.  Then Enter.                                  More ...


  Transaction Scheduler . . ASCH____

  To system file . .  SA28.APPCTP_________________________________


    TP Name  YOURTP__________________________________________________________

    Level . . . . . SYSTEM     System/Group/User

    ID  . . . . . . ________   Group ID or User ID
                               (required for Group or User Level)


  Active Status . . YES_       Yes/No



PF01 = Help     PF03 = Exit     PF12 = Cancel
Command ===>
```

*Figure 10. Panel ICQASE08 - Adding a TP Profile (1 of 3)*

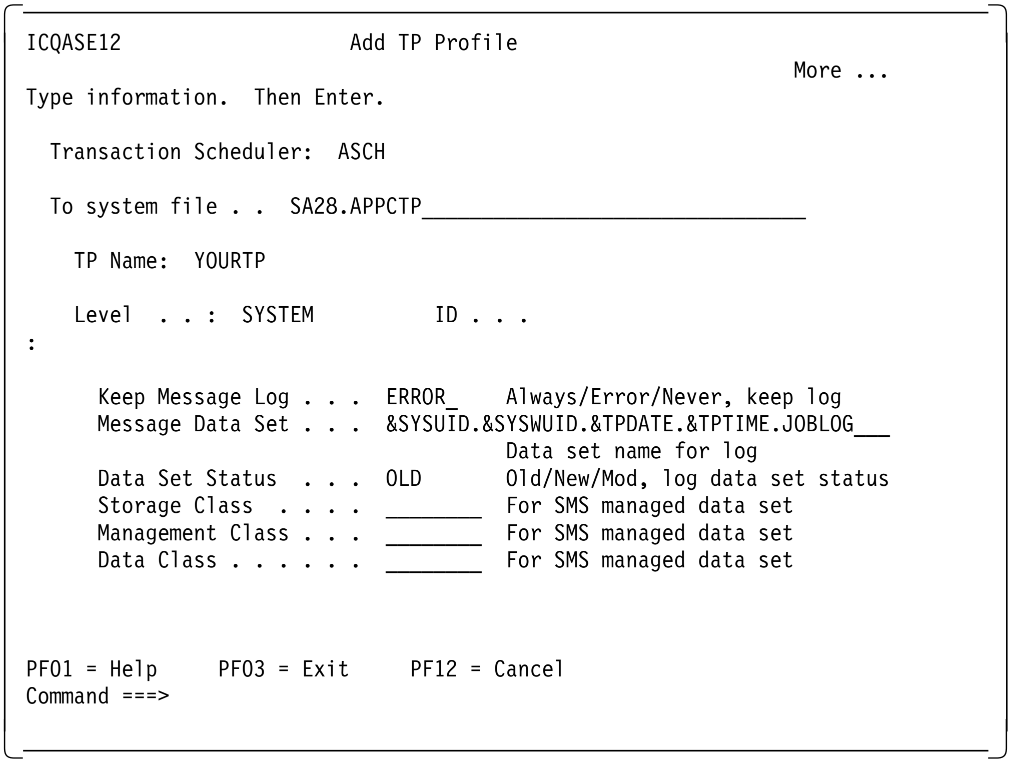
d. On panel ICQASE10, in Figure 11 on page 29, enter STD in the field labelled TP Scheduler Type and press the Enter key.



```
ICQASE10                   Add TP Profile
                                                              More ...
Type information.  Then Enter.

  Transaction Scheduler:  ASCH

  To system file . .  SA28.APPCTP________________________________

    TP Name:  YOURTP

    Level  . . :  SYSTEM         ID . . .
:

      Tailor Sysout  . . . .  NO__       Yes/No, perform sysout tailoring
      Tailor Account . . . .  NO__       Yes/No, perform account tailoring
      Scheduler Class  . . .  ________   Scheduled class when attached
                                         (enter "*" for list of classes)
      TP Schedule Type . . .  STD__      Std (Standard) / Multi (Multitrans)
      Generic Userid . . . .  ________   TP "owner", required for Multitrans




PF01 = Help     PF03 = Exit     PF12 = Cancel
Command ===>
```

*Figure 11. Panel ICQASE10 - Adding a TP Profile (2 of 3)*

e. Press Enter on panel ICQASE12, shown in Figure 12.

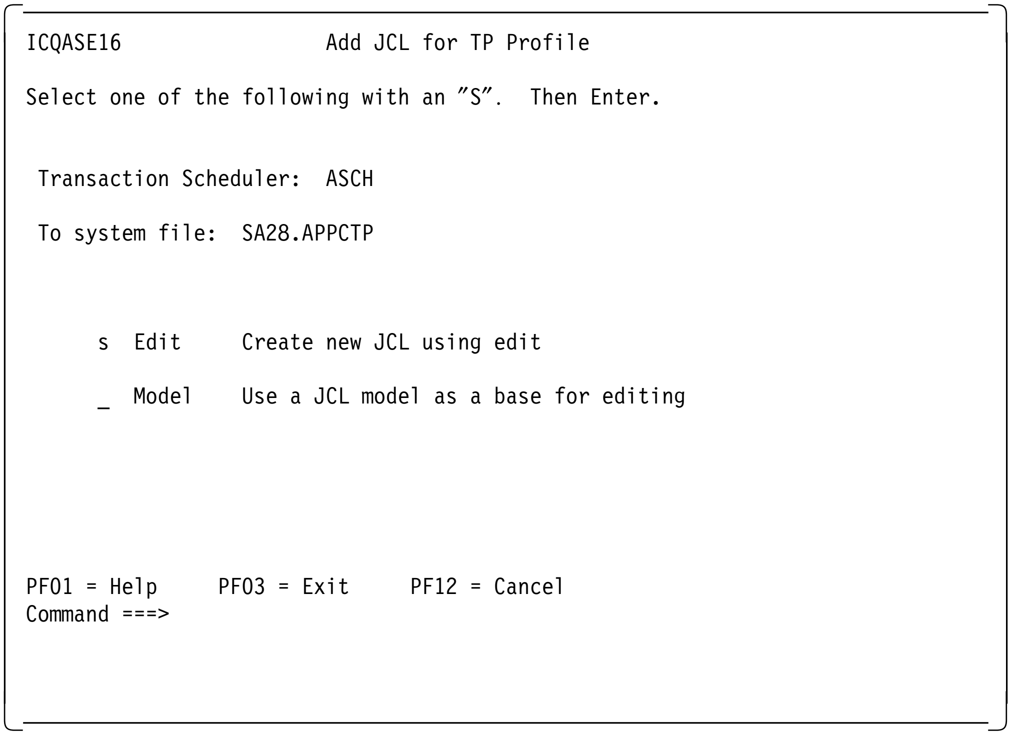


```
ICQASE12                   Add TP Profile
                                                              More ...
Type information.  Then Enter.

  Transaction Scheduler:  ASCH

  To system file . .  SA28.APPCTP________________________________

    TP Name:  YOURTP

    Level  . . :  SYSTEM         ID . . .
:

      Keep Message Log . . .  ERROR_    Always/Error/Never, keep log
      Message Data Set . . .  &SYSUID.&SYSWUID.&TPDATE.&TPTIME.JOBLOG___
                                        Data set name for log
      Data Set Status  . . .  OLD       Old/New/Mod, log data set status
      Storage Class  . . . .  ________  For SMS managed data set
      Management Class . . .  ________  For SMS managed data set
      Data Class . . . . . .  ________  For SMS managed data set



PF01 = Help     PF03 = Exit     PF12 = Cancel
Command ===>
```

*Figure 12. Panel ICQASE12 - Adding a TP Profile (3 of 3)*

f. On panel ICQASE16, in Figure 13 on page 30, select the Edit option.

**Note:** This step is *not required* by the ISPF Client/Server client. It is required by the HLAPI/2 and the CICS clients *only*.



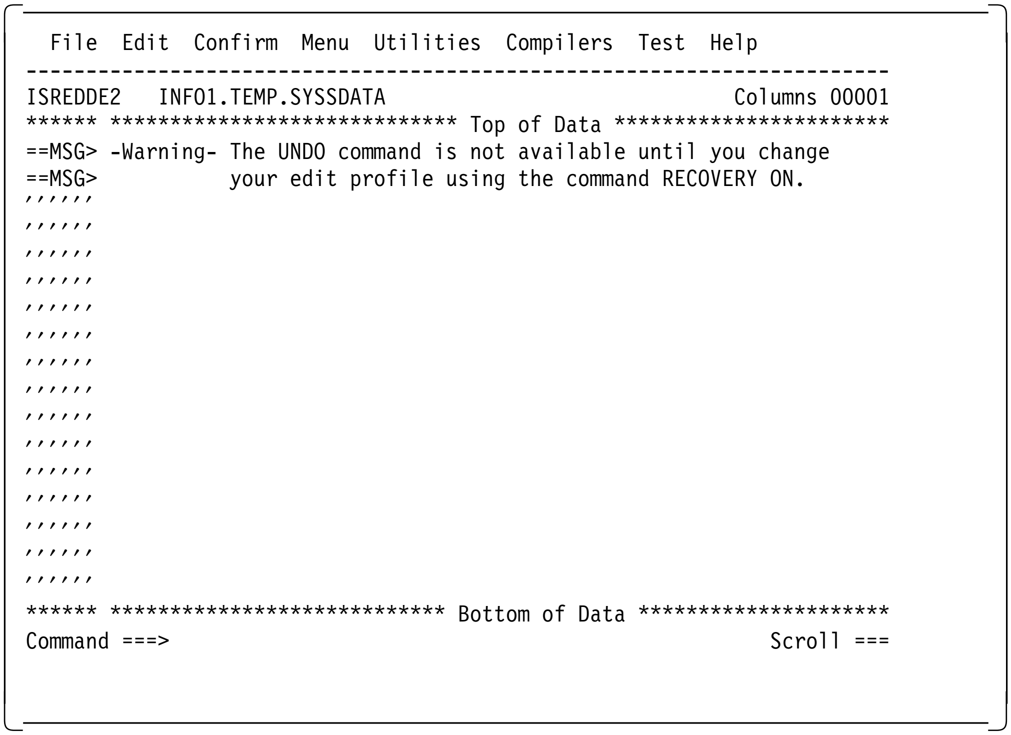
```
ICQASE16                 Add JCL for TP Profile

Select one of the following with an "S".  Then Enter.


 Transaction Scheduler:  ASCH

 To system file:  SA28.APPCTP



       s  Edit     Create new JCL using edit

       _  Model    Use a JCL model as a base for editing





PF01 = Help     PF03 = Exit     PF12 = Cancel
Command ===>
```

*Figure 13. Panel ICQASE16 - Add JCL for TP Profile (HLAPI)*

g. This takes you into ISPF edit on a temporary data set, shown in Figure 14 on page 31, because APPC/MVS expects to be given JCL to submit when ASCH schedules a task.

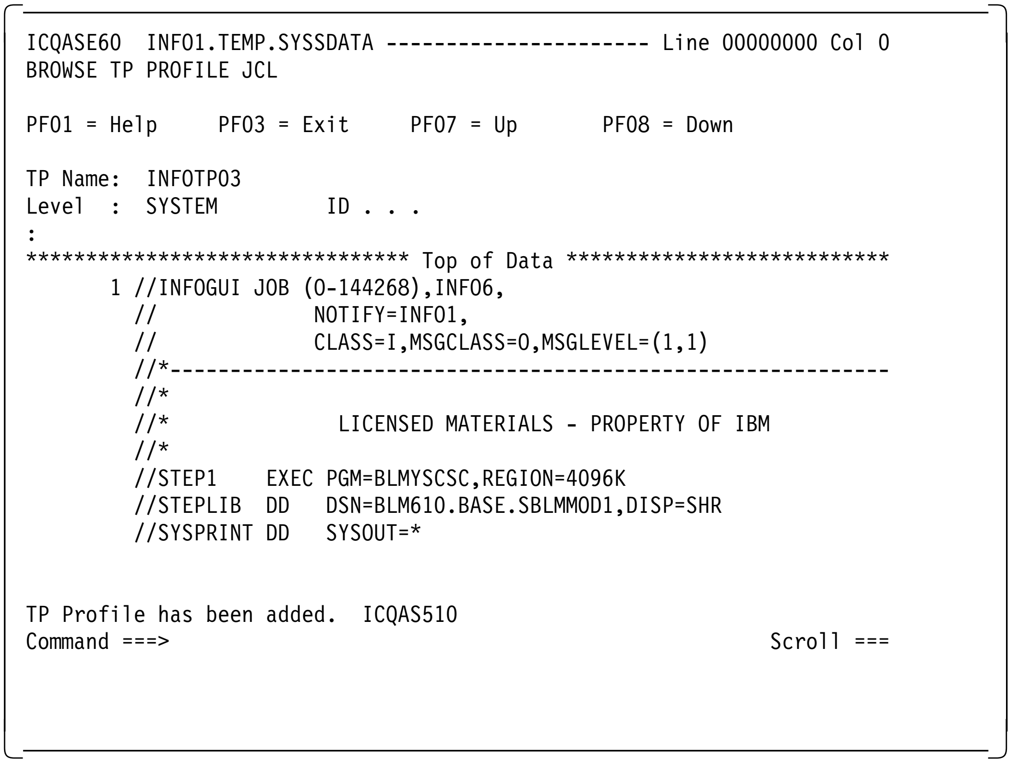
   **Note:** This step is *not required* by the ISPF Client/Server client. It is required by the HLAPI/2 and the CICS clients *only*.

h. Use the JCL in Figure 15 on page 31 as an example of this. Ensure the JCL is coded in uppercase, and press PF3 when you finish entering the JCL. You will be taken back to panel ICQASE16, and be presented with message ICQAS591 inviting you to press Enter. When you press Enter, you will be taken to panel ICQASE60, show in Figure 15 on page 31.



```
  File  Edit  Confirm  Menu  Utilities  Compilers  Test  Help
-------------------------------------------------------------------------
ISREDDE2   INFO1.TEMP.SYSSDATA                             Columns 00001
****** ***************************** Top of Data **********************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
****** **************************** Bottom of Data ********************
Command ===>                                                  Scroll ===
```

*Figure 14. Panel ISREDDE2 - ISPF Edit of Scheduler JCL (HLAPI)*



```
ICQASE60  INFO1.TEMP.SYSSDATA ---------------------- Line 00000000 Col 0
BROWSE TP PROFILE JCL

PF01 = Help     PF03 = Exit     PF07 = Up       PF08 = Down

TP Name:  INFOTP03
Level  :  SYSTEM         ID . . .
:
******************************** Top of Data **************************
       1 //INFOGUI JOB (O-144268),INF06,
         //             NOTIFY=INFO1,
         //             CLASS=I,MSGCLASS=O,MSGLEVEL=(1,1)
         //*----------------------------------------------------------
         //*
         //*              LICENSED MATERIALS - PROPERTY OF IBM
         //*
         //STEP1    EXEC PGM=BLMYSCSC,REGION=4096K
         //STEPLIB  DD   DSN=BLM610.BASE.SBLMMOD1,DISP=SHR
         //SYSPRINT DD   SYSOUT=*


TP Profile has been added.  ICQAS510
Command ===>                                                  Scroll ===
```

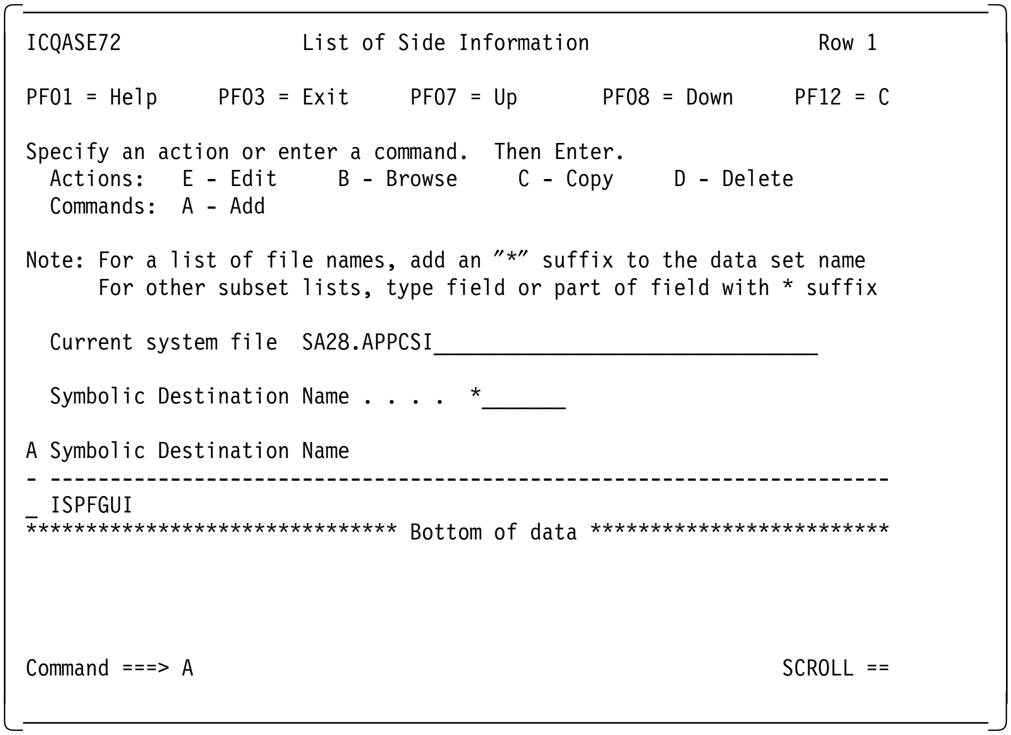
*Figure 15. Panel ICQASE60 - Acknowledgement of New Addition (HLAPI)*

i. Press PF3 on panel ICQASE60 in Figure 15. This takes you back to panel ICQASE04, as shown in Figure 9 on page 27. Press PF3 on that panel to get back to panel ICQASE02, as shown in Figure 8 on page 27.

4. Define side information in the APPC/MVS APPCSI file.

a. Select Side Information Administration on panel ICQASE02 in Figure 8 on page 27 and then press the Enter key.

b. Enter A on the command line of panel ICQASE72 (see Figure 16) to create a new side Symbolic Destination Table entry.



```
ICQASE72               List of Side Information                   Row 1

PF01 = Help     PF03 = Exit     PF07 = Up       PF08 = Down     PF12 = C

Specify an action or enter a command.  Then Enter.
  Actions:   E - Edit     B - Browse     C - Copy     D - Delete
  Commands:  A - Add

Note: For a list of file names, add an "*" suffix to the data set name
      For other subset lists, type field or part of field with * suffix

  Current system file  SA28.APPCSI________________________________

  Symbolic Destination Name . . . .  *_______

A Symbolic Destination Name
- ----------------------------------------------------------------------
_ ISPFGUI
******************************* Bottom of data *************************



Command ===> A                                                SCROLL ==
```

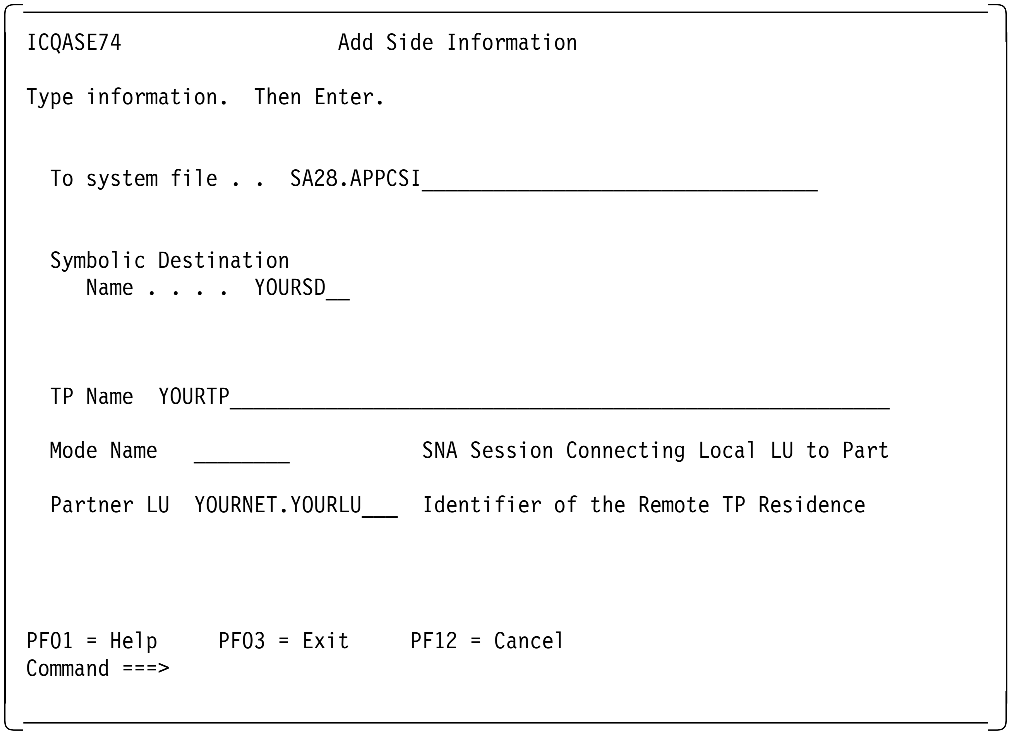
*Figure 16. Panel ICQASE72 - The Side Information Panel*

c. On panel ICQASE74, you will need to choose a symbolic destination name of your own to enter here. There are no restrictions (other than length) on what can be entered, but the important point to note is that this matches the side information you refer to in the workstation definitions later. Enter the TPNAME you defined earlier on the TP Name field. Be careful with the field labelled Partner LU. The value you enter here must match the fully qualified network name of the LU that has been defined as available for LU6.2 on the workstation. This is the LU that has been defined in VTAM as having LOCADDR 0, and should not be confused with any LUs that the workstation may have available for 3270 emulated sessions.

When you have entered data in the fields, press the Enter key to continue.

Record your Symbolic Destination Name below for future reference:

________________________________________



```
ICQASE74                 Add Side Information

Type information.  Then Enter.


  To system file . .  SA28.APPCSI_________________________________


  Symbolic Destination
     Name . . . .  YOURSD__



  TP Name  YOURTP_______________________________________________________

  Mode Name   ________           SNA Session Connecting Local LU to Part

  Partner LU  YOURNET.YOURLU___  Identifier of the Remote TP Residence




PF01 = Help     PF03 = Exit     PF12 = Cancel
Command ===>
```

*Figure 17. Panel ICQASE74 - Adding Side Information*

   d. If you enter valid data, then you will receive message ICQAS410 and be taken back to panel ICQASSE74, as shown in Figure 17. If you press PF3 on that panel, you will be taken back to the APPC/MVS administration menu, panel ICQASE02, as shown in Figure 8 on page 27. Then press PF3 to leave APPC/MVS

      Your side information has been added.

5. You will need to have APPC and ASCH started tasks defined. You can define them as shown in Figure 18 and Figure 19 and Figure 20 on page 34 in your PARMLIB and PROCLIB.

```
//APPC PROC APPC=28
//APPC EXEC PGM=ATBINITM,PARM='APPC=&APPC',REGION=OK
```

*Figure 18. SYS1.PROCLIB: APPC*

```
//ASCH PROC ASCH=28
//ASCH EXEC PGM=ASBSCHIN,PARM='ASCH=&ASCH',REGION=OK
```

*Figure 19. SYS1.PROCLIB: ASCH*

6. Use SDSF to enter the operator command `S APPC,SUB=MSTR`
7. Use SDSF to enter the operator command `S ASCH,SUB=MSTR`

Please refer to *Jumping out of the Box: New Interfaces and Features of Information/Management Version 6 Release 1*, GG24-4325 which describes how to download the ISPF Client/Server (ISPFCS) code to the workstation for the graphical user interface, and how to install HLAPI/2.

```
/*****************************************************************/
/*                                                               */
/*                                                               */
/*   Note:                                                       */
/*                                                               */
/*   1. The VSAM data set specified on the TPDATA keyword         */
/*      must be already defined to the system, otherwise an       */
/*      error message will be issued.                            */
/*                                                               */
/*   2. If this LU is intended to handle network traffic,        */
/*      then the LU name MVSLU01 must correspond to the          */
/*      ACBNAME operand of an application definition (APPL)      */
/*      statement in the SYS1.VTAMLST system library and         */
/*      the ACBNAME & APPL name must match.                      */
/*                                                               */
/*****************************************************************/
  LUADD ACBNAME(RASAZ)       /* Add local LU RASAZ to the          */
                             /* APPC/MVS configuration             */
        SCHED(ASCH)          /* Specify that the APPC/MVS          */
                             /* transaction scheduler is associated */
                             /* with this LU name                  */
        BASE                 /* Designate this LU as the base LU   */
        TPDATA(SA28.APPCTP)  /*                                    */
                             /* RISC.APPCTP is the permanent       */
                             /* repository for the TP profiles     */
                             /* for this LU                        */
        TPLEVEL(USER)        /* Specify the search order for TP    */
                             /* profiles as :                      */
                             /* 1. TP profiles associated with     */
                             /*    a specific user                 */
                             /* 2. TP profiles associated with     */
                             /*    a group of users                */
                             /* 3. TP profiles associated with     */
                             /*    all users of the LU name        */
```

*Figure 20. SYS1.PARMLIB: APPCxx*

## 5.3 Using the ISPF Client/Server with TCP/IP

To run the ISPF Client/Server with TCP/IP, you need to satisfy the following pre requisites:

1. IBM TCP/IP for MVS Version 2.2 needs to be installed on an MVS/ESA system.

2. A TCP/IP started task needs to be active on that system.

   If the started task is not called TCP/IP, or if the ISPDFLTS module was not customized for TCP/IP then you need to follow the steps outlined in 5.3.1, “Customizing ISPF for TCP/IP” on page 35.

3. You need to follow a series of steps on each workstation that you wish to run the TCP/IP GUI. They are detailed in *Jumping out of the Box: New Interfaces and Features of Information/Management Version 6 Release 1*, GG24-4325.

### 5.3.1 Customizing ISPF for TCP/IP

If your TCP/IP started task has a different name from the default value of TCPIP, or if the ISPDFLTS module was not customized for TCP/IP then you will need to make the following modifications to your ISPF environment. For further details see *ISPF Planning and Customizing Version 4 Release 1 for MVS*, SC34-4443.

1. Update ISPDFLTS module. An example is shown in Figure 21.

```
*    COPYRIGHT =
*       5684-054 COPYRIGHT (C) IBM CORP 1987,  1994
*       LICENSED MATERIAL - PROGRAM PROPERTY OF IBM
*       REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083
*
*  THE FOLLOWING MACRO INVOCATION WILL GENERATE AN ISPF DEFAULTS
*  TABLE, ISPDFLTS.
*
*
         ISPMTAIL   TROWDEF=1,CMDSTKSZ=512,EXITS=NO,TCP=TCPIP18
*                                                               C6@MEC
*
```

*Figure 21. ISPDFLTS Source Member*

2. Assemble and link-edit the ISPDFLTS module to create a new ISPDFLTS load module. A sample job is shown in Figure 22.

```
//jobcard
//*****************************************************************
//ASM       EXEC PGM=IEV90,REGION=1024K,PARM='NODECK,LOAD'
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSLIB    DD DSN=your.sispmacs,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//SYSLIN    DD DSN=&LOADSET,UNIT=SYSDA,DISP=(MOD,PASS),
//             SPACE=(80,(200,50))
//SYSIN     DD DSN=ISP410.BASE.SAMPJCL(ISPDFLTS),DISP=SHR
//LINK1     EXEC PGM=IEWL,PARM='LIST,XREF,LET,NORENT'
//SYSPRINT  DD SYSOUT=*
//SYSLMOD   DD DSN=your.sispmod1,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(50,10))
//SYSLIN    DD DSN=&LOADSET,DISP=(OLD,DELETE)
//          DD *
            ENTRY ISPDFLTS
            NAME ISPDFLTS(R)
/*
```

*Figure 22. Assembly and Link-Edit of ISPDFLTS*

3. Link-edit ISPMAIN to include the updated ISPDFLTS. A sample job is shown in Figure 23 on page 36.

```
//jobcard
//***********************************************************
//LINK1     EXEC PGM=IEWL,PARM='LIST,XREF,LET,NORENT'
//SYSPRINT  DD SYSOUT=*
//SYSLMOD   DD DSN=your.sispload,DISP=SHR
//SISPLPA   DD DSN=your.sisplpa,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(50,10))
//SYSLIN    DD DSN=your.sispmod1(ISPDFLTS),DISP=SHR
//          DD *
            INCLUDE SISPLPA(ISPMAIN)
            ENTRY ISPMDR
            NAME ISPMAIN(R)
/*
```

*Figure 23. Link-Edit of ISPMAIN*

# Chapter 6. Multisystem Database Access

The Multisystem Database Access feature of Information/Management Version 6 Release 1 allows access to the same Information/Management VSAM data sets from multiple MVS/ESA systems or multiple BLXx procedures in the same MVS/ESA system.

Multiple BLX systems in the same MVS/ESA are not convenient for a production environment, but are useful for test purposes. For that reason their implementation is discussed in this document.

This section covers the steps necessary to enable Multisystem Database Access in single and multiple MVS/ESA system environments.

Performance implications will be covered, but for detailed performance information refer to *Boosting Your Information/Management Version 6 Release 1 Performance*, GG24-4324.

## 6.1 Multisystem Database Access in a Single System

Implementing Multisystem Database Access in a single or in a multi-system environment require the same tasks.

Figure 24 depicts an overview of Multisystem Database Access in a single system environment.

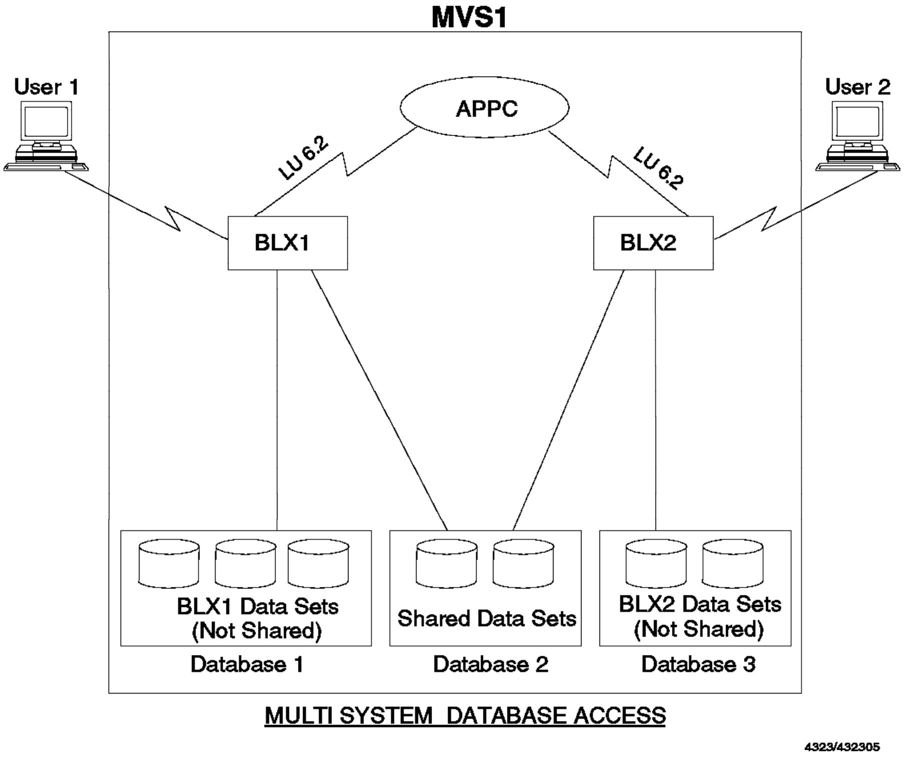


*Figure 24. Multisystem Database Access on One MVS/ESA System*

The above diagram represents the following scenario:

- One MVS/ESA system (MVS1).
- Two BLX-SP procedures (BLX1SH and BLX2SH).
- Two BLX-SP servers address spaces, (BLX1 and BLX2) connected via LU 6.2 connection to APPC.
- The BLXVDEF member for the BLX1 service provider contains two Information/Management databases:

  Database 1 - defined as not shareable

  Database 2 - defined as shareable.

  The same applies to the BLX2 service provider.
- Two session parameter members - BLGSESxx - are available to the BLX1 service provider. One provides access to Database 1 (which is non-shareable). The other provides access to database 2 (which is shareable).
- The APPC connection to both service providers gets invoked only when the shareable database - database 2 - is updated.

To implement database sharing in a single system environment the following steps need to be completed:

1. Link-edit the Multisystem Database Access module.
2. Tailor the Information/Management setup.
3. Customize Advanced Program-to-Program Communication (APPC).
4. Customize APPC/MVS Transaction Scheduler.
5. Set up VTAM parameters and modifications.

### 6.1.1 Link-Edit the Multisystem Database Access Module

To enable Multisystem Database Access, job BLXCSDL needs to be run. Before executing it, verify with the personnel who installed Information/Management whether or not BLXCSDL was run during the installation. The original JCL is found in BLM.V6R1M0.SBLMSAMP. The member name is BLXCSDL. The sample is shown in Figure 25 on page 39.

```
//BLXCSDL  JOB (ACCT),′LINKEDIT  MDA′                                   00010000
//**************************************************************/       00110000
//* THIS JOB IS USED TO LINK-EDIT THE INFORMATION/MANAGEMENT   */       00120000
//* MULTISYSTEM DATABASE ACCESS CODE.  APPC LIBRARY ROUTINES   */       00130000
//* MUST BE LINK-EDITED WITH THE FEATURE CODE TO FORM THE      */       00140000
//* LOAD MODULES.  THE FEATURE CODE MUST BE LINK-EDITED USING  */       00150000
//* MVS/ESA VERSION 4 RELEASE 2 OR LATER APPC LIBRARY          */       00160000
//* SUBROUTINES.                                               */       00170000
//*                                                            */       00180000
//*    UPDATE THE FOLLOWING WITH UPPERCASE TEXT:               */       00190000
//*                                                            */       00200000
//*    1)  JOB CARD INFORMATION                                */       00210000
//*    2)  ANY DATA SET INFORMATION WHICH DOES NOT             */       00220000
//*        MATCH THAT USED AT YOUR INSTALLATION                */       00230000
//*                                                            */       00240000
//**************************************************************/       00280000
//*                                                                     00290000
//LINK1   EXEC PGM=IEWL,REGION=1024K,                                   00300000
//       PARM=(RENT,REUS,                                               00310000
//             XREF,LET,LIST,MAP,′SIZE=(512K,128K)′,NCAL)               00320000
//SYSPRINT DD SYSOUT=*                                                  00330000
//SYSUT1   DD UNIT=SYSDA,SPACE=(TRK,(10,5))                             00340000
//SYSLMOD  DD DSN=BLM.V6R1M0.SBLMMOD1,DISP=OLD  /* LOAD LIBRARY  */     00360000
//SBLMTXT1 DD DSN=BLM.V6R1M0.SBLMTXT1,DISP=SHR  /* FEATURE CODE  */     00380000
//APPCLIB  DD DSN=SYS1.CSSLIB,DISP=SHR          /* APPC MODS     */     00400000
//SYSLIN   DD *,DLM=′$$′                                                00410000
  INCLUDE APPCLIB(ATBPBI)                                               00420000
  INCLUDE SBLMTXT1(BLXCBIAR)                                            00430000
  INCLUDE SBLMTXT1(BLXCCRBI)                                            00440000
  INCLUDE SBLMTXT1(BLXCCSAK)                                            00450000
  INCLUDE SBLMTXT1(BLXCBIAS)                                            00460000
 MODE  AMODE(31),RMODE(ANY)                                             00470000
 SETCODE AC(1)                                                          00480000
 ENTRY BLXCBIAS                                                         00490000
 NAME BLXCBIAS(R)                                                       00500000
  INCLUDE SBLMTXT1(BLXCCNVI)                                            00510000
  INCLUDE SBLMTXT1(BLXCCSBI)                                            00520000
  INCLUDE SBLMTXT1(BLXCCRAK)                                            00530000
  INCLUDE SBLMTXT1(BLXCCNVT)                                            00540000
  INCLUDE SBLMTXT1(BLXCBITT)                                            00550000
  INCLUDE SBLMTXT1(BLXCBITR)                                            00560000
  INCLUDE APPCLIB(ATBPBI)                                               00570000
  INCLUDE SBLMTXT1(BLXCBIST)                                            00580000
 MODE  AMODE(31),RMODE(ANY)                                             00590000
 ENTRY BLXCBIST                                                         00600000
 NAME BLXCBIST(R)                                                       00610000
$$                                                                      00620000
/*
```

*Figure 25. Sample JCL to Link-Edit Multisystem Database Access*

## 6.1.2 Information/Management Setup

Customizing Information/Management for Multisystem Database Access involves setting up the environment for MVS1 as shown in Figure 24 on page 37. The following needs to be done:

1. Set up two started tasks (BLX1SH and BLX2SH) in SYS1.PROCLIB with the sample JCL shown in Figure 26 on page 40.

SYS1.PROCLIB(BLX1SH)

```
//BLX1SH   PROC PRM=SH
//BLXSPCAS EXEC PGM=BLXSSP00,REGION=6M,TIME=1440,PARM=&PRM
//STEPLIB  DD   DSN=BLM.V6R1M0.SBLMMOD1       APF AUTHORISED
//BLXPRM   DD   DISP=SHR,BLM.V6R1M0.SBLMSAMP  INFO BLX-SP PARMS
```

SYS1.PROCLIB(BLX2SH)

```
//BLX2SH   PROC PRM=SH
//BLXSPCAS EXEC PGM=BLXSSP00,REGION=6M,TIME=1440,PARM=&PRM
//STEPLIB  DD   DSN=YOUR.BLX2MOD.LIBRARY      APF AUTHORISED
//         DD   DSN=BLM.V6R1M0.SBLMMOD1       APF AUTHORISED
//BLXPRM   DD   DISP=SHR,BLM.V6R1M0.SBLMSAMP  INFO BLX-SP PARMS
```

*Figure 26. PROCLIB Changes*

YOUR.BLX2MOD.LIBRARY indicated in the second procedure of Figure 26 is an additional library, which is APF authorized and should contain at least the BLXSSINM module.

The BLXSSINM module has a pointer to BLX2 which is your second Information/Management started task. The source of the module is in BLM.V6R1M0.SBLMSRC1 member name BLXSSINM. The abbreviated JCL and the source are shown in Figure 27 on page 41.

```
//ASMBLX2  JOB  (ACCT),'ASM AND LINK BLX2'
//ASMBLX    EXEC PGM=IEV90,REGION=1024K,PARM='NODECK,LOAD'
//SYSPRINT  DD SYSOUT=*
//SYSUT1    DD UNIT=SYSDA,SPACE=(CYL,(3,1))
//SYSLIB    DD DSN=BLM.V6R1M0.SBLMMACS,DISP=SHR
//          DD DSN=SYS1.MACLIB,DISP=SHR
//SYSLIN    DD DSN=&LOADSET,UNIT=SYSDA,DISP=(MOD,PASS),
//             SPACE=(80,(200,50))
//SYSIN     DD *,DLM='$$'
*/* BLXSSINM SOURCE MODULE *
BLXSSINM CSECT
         DC    CL4'BLX2'    Name of your second Central address space
         END
$$
/*
//LINKBLX   EXEC PGM=IEWL,PARM='LIST,XREF,RENT,NCAL,MAP,REUS',
//          COND=(0,NE)
//SYSPRINT  DD SYSOUT=*
//SYSLMOD   DD DSN=YOUR.BLX2MOD.LIBRARY,DISP=SHR
//SYSUT1    DD UNIT=SYSDA,SPACE=(TRK,(50,10))
//SYSLIN    DD DSN=&LOADSET,DISP=(OLD,DELETE)
//SYSIN     DD *,DLM=$$
           ENTRY BLXSSINM
           NAME BLXSSINM
$$
/*
```

*Figure 27. Sample JCL with the Source for the BLXSSINM Module*

**Note** Please refer to the chapter on *Setting Up Your BLX-SP* in the *Planning and Installation Guide*, for more information on multiple servers.

2. Changes to SYS1.PARMLIB

   - Include BLX2 subsystem name in the subsystem name table, IEFSSNxx.
   - Include BLXCBIAS in the program properties table, SCHEDxx, as shown in Figure 28.

```
 /*******************************************************************/
 /*                                                                 */
 /*          PPT FOR INFO V6 MULTISYSTEM DATABASE ACCESS            */
 /*                                                                 */
 /*******************************************************************/
   PPT PGMNAME(BLXCBIAS)      /* PROGRAM NAME INFO V6               */
       NOSWAP                 /* NON-SWAPPABLE                      */
```

*Figure 28. SCHEDxx Member of SYS1.PARMLIB*

3. Allocate the Information/Management shared data sets. All data sets that will be shared must be defined with SHROPTIONS (3 3), except for the Log data set, which is defined with SHROPTIONS (4 4). These allocations are made so that multiple reads and writes can happen simultaneously.

4. Set up your BLXVDEFx for your local shared resources and non shared resources. This should include your shared data sets. The data sets should be specified with SHARE=YES as indicated in Figure 29 on page 42.

   DSNLABEL BLXDSN DSN=Info.data.set,LSR=(x,x),SHARE=YES.

```
BLXVDEFS CSECT
         SPACE 2
***********************************************************************
*     DEFINE NON-SHARED RESOURCES TO BLX-SP                           *
***********************************************************************
         BLXNSR PLACES=20                 TWENTY PLACES FOR NSR.
         SPACE 2
***********************************************************************
*     DEFINE VSAM DATA SETS TO BLX-SP AND CONNECT VSAM RESOURCES      *
***********************************************************************
*     DEFINE DATA SETS USING LOCAL SHARED RESOURCES (LSR)
BASEPNLS BLXDSN DSN=BLM.V6R1M0.IBMPNLS,LSR=(2,2),SHARE=YES
DICT     BLXDSN DSN=BLM.V6R1M0.DICT,LSR=(2,2),SHARE=YES
MGMTSDDS BLXDSN DSN=BLM.V6R1M0.SDDS,LSR=(0,0),SHARE=YES
MGMTINDX BLXDSN DSN=BLM.V6R1M0.SDIDS,LSR=(1,1),SHARE=YES
*
*     DEFINE DATA SETS USING NON-SHARED RESOURCES (NSR)
MGMTLOG  BLXDSN  DSN=BLM.V6R1M0.SDLDS,DSORG=NUM,SHARE=YES
         SPACE 2
*     GENERATE VSAM DEFINITION FOR BLX-SP                             *
         BLXGEN                           GENERATE DEFINITION
         SPACE 2
*        RESOURCE POOL ZERO (SDDS DATA BUFFER POOLS)
LSRD0  BLDVRP  BUFFERS=(4096(31)),      SDDS DATA BUFFERS              X
               KEYLEN=7,                POOL USED BY SDDS              X
               STRNO=30,                MAINTAIN 30 POSITIONS          X
               SHRPOOL=0,               IDENTIFY POOL                  X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY              X
               RMODE31=ALL,MODE=24,MF=L
*
*        RESOURCE POOL ZERO (SDDS INDEX BUFFER POOLS)
*
LSRI0  BLDVRP  BUFFERS=(1536(53)),      SDDS INDEX BUFFERS AND         X
               SHRPOOL=0,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,MODE=24,MF=L
         SPACE 2
***********************************************************************
*     RESOURCE POOL ONE (FOR PRODUCTION DATABASE SDIDS)               *
LSRD1  BLDVRP  BUFFERS=(2048(500)),     SDIDS DATA BUFFERS + PAD       X
               KEYLEN=32,               POOL USED BY SDIDS             X
               STRNO=45,                MAINTAIN 45 POSITIONS          X
               SHRPOOL=1,               IDENTIFY POOL                  X
               TYPE=(LSR,DATA),         DATA BUFFERS ONLY              X
               RMODE31=ALL,MODE=24,MF=L
*
*        RESOURCE POOL ONE (SDIDS INDEX BUFFER POOLS)
LSRI1  BLDVRP  BUFFERS=(2560(32)),      SDIDS INDEX BUFFERS            X
               SHRPOOL=1,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,MODE=24,MF=L
         SPACE 2
*     RESOURCE POOL TWO  (PANEL/DICTIONARY DATA SETS)                 *
LSRD2  BLDVRP  BUFFERS=(4096(36)),      DICT/PANEL DATA BUFFERS        X
               KEYLEN=10,               POOL USED BY DICT/PANELS       X
               SHRPOOL=2,               POOL IDENTIFIER                X
               STRNO=35,                COMBINED POSITIONS REQUIRED    X
               TYPE=(LSR,DATA),         INDEX AND DATA POOL            X
               RMODE31=ALL,MODE=24,MF=L
*        RESOURCE POOL TWO (PANEL/DICTIONARY INDEX BUFFER POOLS)
LSRI2  BLDVRP  BUFFERS=(1536(30)),      DICT/PANEL INDEX BUFFERS       X
               SHRPOOL=2,               POOL IDENTIFIER                X
               TYPE=(LSR,INDEX),        INDEX BUFFERS ONLY             X
               RMODE31=ALL,MODE=24,MF=L
         SPACE 2
         END   BLXVDEFS
```

*Figure 29. Sample BLXVDEF Source Code*

The above source can contain other data sets that are not shareable between the Information/Management systems. A different BLGSESxx member can then be defined to access those data sets. Refer to Figure 24 on page 37 for a schematic presentation.

5. Set up your BLGSESxx member with the shared data sets, and specify CAS=1 or 2 to indicate which address space should be used.

   Both BLXVDEF and BLGSESxx must be assembled and link-edited into a load library accessible through a STEPLIB or Link list concatenation.

6. Coding the sample parameter member for the BLX-SP address spaces is the next step in customizing Information/Management. The source shown in Figure 30 is the sample BLX1SH member that can be found in the BLM.V6R1M0.SBLMSAMP data set.

```
/*********************************************************************/
/*                                                                   */
/*         INFO V6.1 BLX-SP OPERATING PARAMETERS                      */
/*         FOR MULTISYSTEM DATABASE ACCESS                             */
/*                                                                   */
/*********************************************************************/

 BLXPRM                             /* SPECIFY BLX-SP PARAMETERS     */

        TRACE=OFF,                  /* DON’T PRODUCE TRACE OUTPUT    */
/*      LOG OPTIONS                                                  */
        LOG=ON,                     /* PRODUCE LOG INFORMATION       */
        LOGSYSOUT=A,                /* JES SYSOUT CLASS FOR LOG DS   */
        LOGLINES=0,                 /* MAX # OF LINES IN A LOG DS    */

     /*   BLX-SP SHUT DOWN OPTIONS                              */

        SHUTDOWNWT=00050000,        /* SHUTDOWN WAIT TIME HHMMSSTH   */
        SHUTDOWNTFY=00050000,       /* SHUTDOWN NOTIFY WT HHMMSSTH   */

     /*   BLX-SP MESSAGE ROUTING OPTIONS                        */

        WRITEOPER=1,                /* DEFAULT WTO ROUTING CODE      */

     /***********************************************************/
     /*                                                         */
     /*   SYMBOLIC DESTINATION NAMES (FOR SHARING DATASETS)     */
     /*                                                         */
     /***********************************************************/

        DESTNAMES=(SYSTEM1,SYSTEM2),
                                    /* SYSTEM1 is the local system   */
                                    /* SYSTEM2 & SYSTEM3 represent   */
                                    /* the systems that share data   */
                                    /* sets with SYSTEM1             */

     /*   BLX-SP VSAM RELATED OPTIONS                           */

        VSAMRESOURCES=BLXVDEFS;     /* BLXDSN, BLXNSR, BLXGEN AND    */
                                    /* THE VSAM BLDVRP MACROS        */
```

*Figure 30. Sample BLX1SH Parameters*

Keep in mind when setting the DESTNAMES parameter that the first system specified is treated as the local system and the second and third ones are your remote systems. For this example, SYSTEM1 is the local and SYSTEM2 is the remote system.

SYSTEM2 must be specified first in the BLX2SH parameters because it is the local system for the BLX2SH procedure. The DESTNAMES for the BLX2SH procedure will be as follows:

```
  DESTNAMES=(SYSTEM2,SYSTEM1)
```

Please record the names. They are required for the Advanced Program-to-Program Communication (APPC) and APPC/MVS Transaction Scheduler setups.

## 6.1.3 Advanced Program-to-Program Communication (APPC) Setup

Advanced Program-to-Program Communication (APPC) implements the Systems Network Architecture (SNA) LU 6.2 protocol. It supports communication of programs running in two or more connected systems. The following needs to be done to customize APPC for Multisystem Database Access.

### 6.1.3.1 Transaction Program Profile

The Transaction Program profile (TP) is used for communication between the APPC and the network. In the local Multisystem Database Access environment, each BLX-SP procedure must have an APPC TP profile. The following values must be set on the TP profile variables:

- TPSCHED_EXIT to (ASCH)
- TPSCHED_TYPE to (STANDARD)
- BLXBIAS as the name of the program in the JCL EXEC statement (in the TPADD statement part of the JCL
- Include a STEPLIB DD statement to point to the libraries that contain the load modules used by the BLX-SP that the current profile is associated with. This is not required if the libraries are in the link list concatenation.

  Please note that the STEPLIB library concatenation in the TP profile must be the same as the STEPLIB used in the BLX-SP cataloged procedure and the Information/Management TSO logon procedure.

The following are optional but recommended:

- The TP profile should be assigned its own class. See APPC/MVS Transaction Scheduler considerations in the next section.
- GROUPID is strongly recommended, but SYSTEM can also be used. The problem with the latter is security. It avoids verification on the user ID and the RACF group of the program accessing this TP, while GROUPID enforces verification. A sample job to add a TP profile to APPC is provided in Figure 31 on page 45. The actual job is found in BLM.V6R1M0.SBLMSAMP(BLXTPADD).

```
//TPADD1   JOB (ACCT),’ADD TP PROFILE’
//*************************************************************
//* THE FOLLOWING DEFAULTS ARE ASSUMED:
//*        I) TYPRUN=RUN
//*       II) TPSCHED_EXIT(ASCH)
//*************************************************************
//STEP1    EXEC PGM=ATBSDFMU
//SYSSDLIB  DD DSN=SYS1.APPCTP,DISP=SHR
//SYSSDOUT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DATA,DLM=XX
TPADD
   TPNAME(SYS1TP)
   GROUPID(INFOGRP)
   TPSCHED_DELIMITER(##)
      CLASS(SHR)
      TPSCHED_TYPE(STANDARD)
      JCL_DELIMITER(END_OF_JCL)
//***************************************************************//
//*  MODIFY JOBCARD INFO TO SUIT INSTALLATION REQUIREMENTS      *//
//***************************************************************//
//***************************************************************/
//BIAS001 JOB  (ACCT),’APPC JOB INFO’
//BIREQ    EXEC PGM=BLXCBIAS
//STEPLIB   DD   DISP=SHR,DSN=BLM.V6R1M0.SBLMMOD1
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD DUMMY
//SYSTSPRT  DD SYSOUT=*
//SYSIN     DD DUMMY
END_OF_JCL
   KEEP_MESSAGE_LOG(ERROR)
##
XX
/*
```

*Figure 31. Sample JCL to Set Up the TP Profile*

Please note that the JOBNAME, BIAS001 in the above example, will be the name that is used in the BLX-SP procedure to identify the user ID accessing the database.

The sample for adding the TP profile for SYSTEM two is shown in Figure 32 on page 46.

```
//TPADD2   JOB (ACCT),'ADD TP PROFILE'
//*********************************************************
//* THE FOLLOWING DEFAULTS ARE ASSUMED:
//*        I) TYPRUN=RUN
//*       II) TPSCHED_EXIT(ASCH)
//*********************************************************
//STEP1    EXEC PGM=ATBSDFMU
//SYSSDLIB  DD DSN=SYS1.APPCTP,DISP=SHR
//SYSSDOUT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DATA,DLM=XX
TPADD
   TPNAME(SYS2TP)
   GROUPID(INFOGRP)
   TPSCHED_DELIMITER(##)
      CLASS(SHR)
      TPSCHED_TYPE(STANDARD)
      JCL_DELIMITER(END_OF_JCL)
//************************************************************//
//*  MODIFY JOBCARD INFO TO SUIT INSTALLATION REQUIREMENTS      *//
//************************************************************//
//BIAS002 JOB  (ACCT),'APPC JOB INFO'
//BIREQ    EXEC PGM=BLXCBIAS
//STEPLIB   DD   DISP=SHR,DSN=YOUR.BLX2MOD.LIBRARY
//          DD   DISP=SHR,DSN=BLM.V6R1M0.SBLMMOD1
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD DUMMY
//SYSTSPRT  DD SYSOUT=*
//SYSIN     DD DUMMY
END_OF_JCL
   KEEP_MESSAGE_LOG(ERROR)
##
XX
/*
```

*Figure 32. Sample JCL to Set Up the TP Profile*

BIAS002 will identify this JOB as a user on the BLX-SP output log.

To verify the customization for Multisystem Database Access, if you have completed the customization successfully, and you initiate the first create on SYSTEM2, then you will get the following message written to the output log for the BLX-SP task associated with SYSTEM1:

```
BLX20135I ASID 0033 TCB 008E1D88 USERID BIAS002 connected to the BLX-SP
```

### 6.1.3.2 Updating the Side Information Table

An entry in the side information table must be created for each BLX-SP address space that uses Multisystem Database Access. The information in this table indicates the connection between the system names, identified in the BLX1SH in the DESTNAMES and the TPNAMES parameters specified in the TP profile. There are additional parameters which will be indicated below.

There are two ways of modifying or adding the side information table information. It can be done via batch or interactively. The sample batch job is shown in Figure 33 on page 47.

```
//SIADD1   JOB (ACCT),'ADD SIDE INFO'
//STEP1    EXEC PGM=ATBSDFMU
//SYSSDLIB  DD DSN=SYS1.APPCSI,DISP=SHR
//SYSSDOUT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *,DLM='$$'
 SIADD
   DESTNAME(SYSTEM1)
   TPNAME(SYS1TP)
   MODENAME(#INTER)
   PARTNER_LU(MVSLU02)
$$
/*
```

*Figure 33. Sample JCL to Add Side Information Table*

The SYSIN information is used as follows;

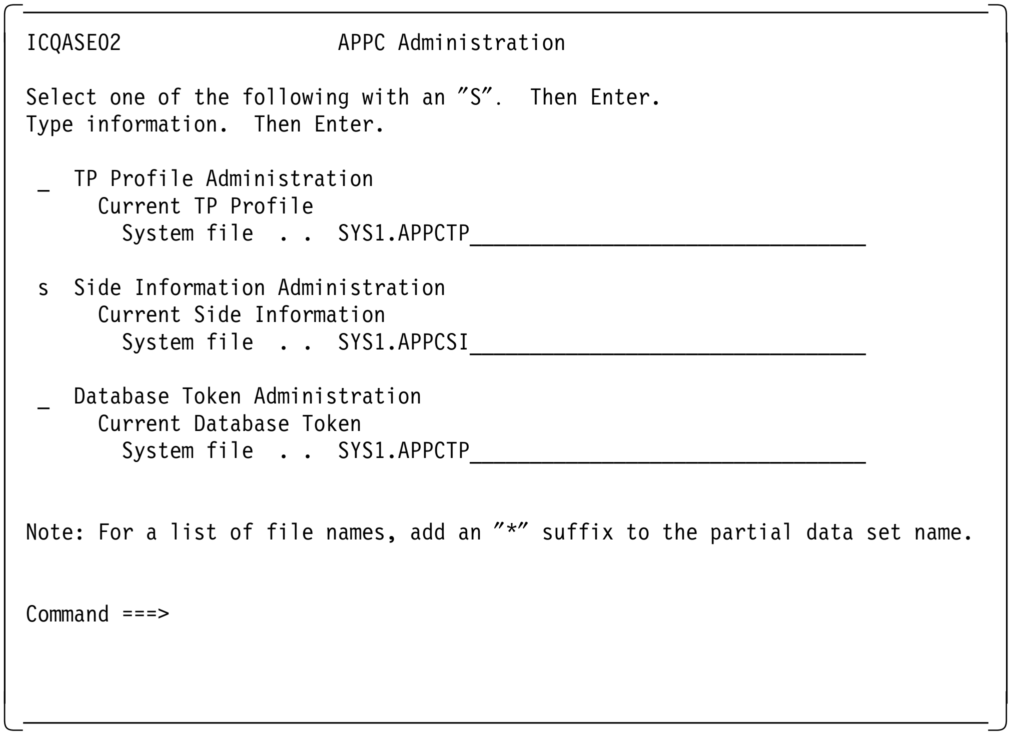
- SIADD - APPC to add symbolic destination names. In the example the symbolic destination name is SYSTEM1.
- TPNAME - the TP profile name associated with the destination name. In the example the TP name is SYS1TP.
- MODENAME - the MODENAME used in VTAM. An installation′s VTAM system programmer would be able to know which MODENAME to use. This value can be left blank and MVS would choose the appropriate MODENAME. The example above uses #INTER.
- PARTNER_LU, the LU that is being communicated with, it must be the name pointed to by ACBNAME in the VTAM Major node. In the example the MVSLU01 is the partner LU. Refer to 6.1.4, "VTAM Considerations" on page 51 for information on how to obtain the LU name.

The side information table for the SYSTEM2 would be as shown in Figure 34. Please note that the symbolic name is SYSTEM2, the TP name is SYS2TP and the partner LU is MVSLU01.

```
//SIADD2   JOB (ACCT),'ADD SIDE INFO'
//STEP1    EXEC PGM=ATBSDFMU
//SYSSDLIB  DD DSN=SYS1.APPCSI,DISP=SHR
//SYSSDOUT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *,DLM='$$'
 SIADD
   DESTNAME(SYSTEM2)
   TPNAME(SYS2TP)
   MODENAME(#INTER)
   PARTNER_LU(MVSLU01)
$$
/*
```

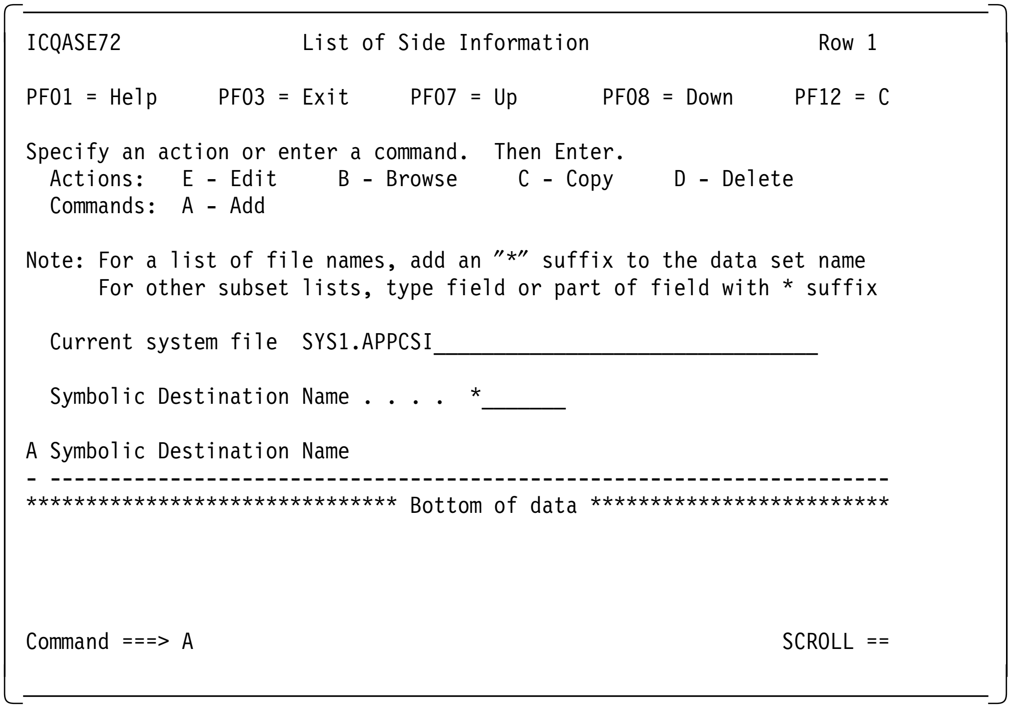
*Figure 34. Sample JCL to Add Side Information Table*

The interactive method involves customizing your ISPF panels to be able to use APPC in interactive mode. This is done through adding option xx,CMD(ICQASRM0) to an ISPF panel. The following panels, displayed in Figure 35 on page 48, Figure 36 on page 48, and in Figure 37 on page 49 with the appropriate selection are shown. Select TP Profile Administration on panel ICQASE02 in Figure 35 on page 48 and press the Enter key.



```
ICQASE02                 APPC Administration

Select one of the following with an "S".  Then Enter.
Type information.  Then Enter.

 _  TP Profile Administration
      Current TP Profile
        System file  . .  SYS1.APPCTP_________________________________

 s  Side Information Administration
      Current Side Information
        System file  . .  SYS1.APPCSI_________________________________

 _  Database Token Administration
      Current Database Token
        System file  . .  SYS1.APPCTP_________________________________


Note: For a list of file names, add an "*" suffix to the partial data set name.


Command ===>
```

Figure 35. Panel ICQASE02 - The APPC/MVS Administration Menu

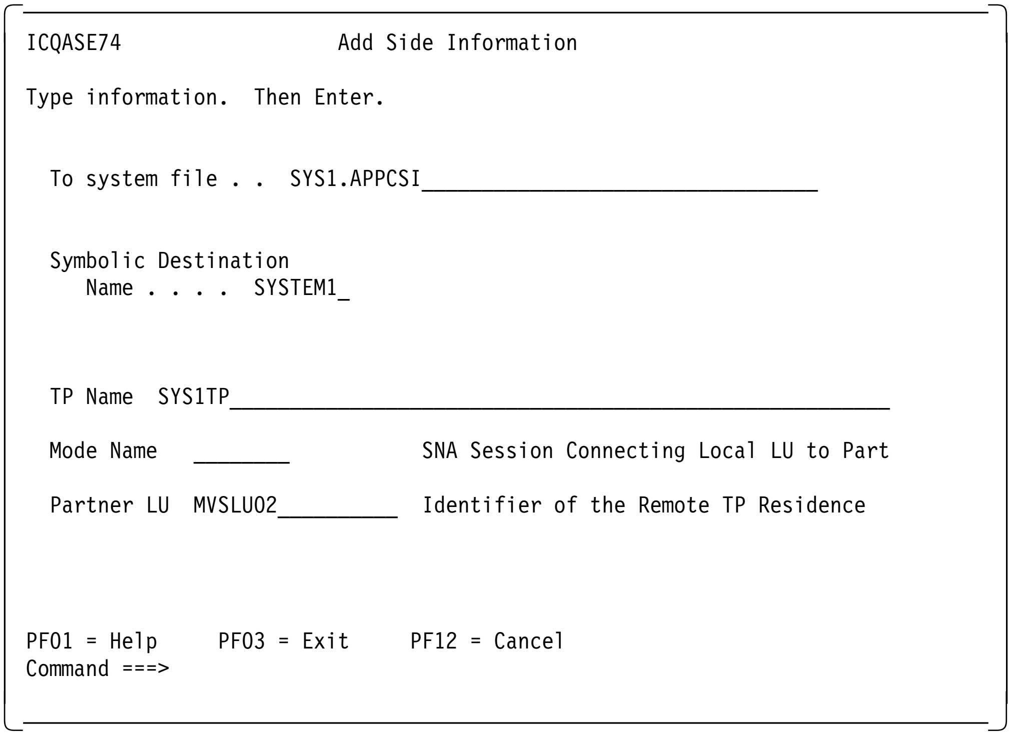


```
ICQASE72              List of Side Information                   Row 1

PF01 = Help     PF03 = Exit     PF07 = Up       PF08 = Down     PF12 = C

Specify an action or enter a command.  Then Enter.
  Actions:   E - Edit     B - Browse     C - Copy     D - Delete
  Commands:  A - Add

Note: For a list of file names, add an "*" suffix to the data set name
      For other subset lists, type field or part of field with * suffix

  Current system file  SYS1.APPCSI________________________________

  Symbolic Destination Name . . . .  *_______

A Symbolic Destination Name
- ----------------------------------------------------------------------
****************************** Bottom of data ************************



Command ===> A                                              SCROLL ==
```

Figure 36. Panel ICQASE72 - The Side Information Panel

On panel ICQASE74, you will need to add the symbolic destination name SYSTEM1. As with the previous method, the value specified here must match the name specified in the BLX1SH in your source library. The values entered in the figure are for SYSTEM1.

When you have entered data in the fields, press the Enter key to continue.



```
ICQASE74                  Add Side Information

Type information.  Then Enter.


  To system file . .  SYS1.APPCSI_________________________________


  Symbolic Destination
     Name . . . .  SYSTEM1_



  TP Name  SYS1TP_______________________________________________________

  Mode Name   ________           SNA Session Connecting Local LU to Part

  Partner LU  MVSLU02__________  Identifier of the Remote TP Residence




PF01 = Help     PF03 = Exit     PF12 = Cancel
Command ===>
```

*Figure 37. Panel ICQASE74 - Adding Side Information*

ON entering the data you will receive message ICQAS410 and be taken back to panel ICQASSE74, as shown in Figure 36 on page 48. If you press PF3 on that panel, you will be taken back to the APPC/MVS administration menu, panel ICQASE02, as shown in Figure 35 on page 48. Then press PF3 to leave APPC/MVS.

Your side information for SYSTEM1 has been added.

The data should be changed as follows for SYSTEM2:

```
Symbolic Destination name: SYSTEM2
TP Name: SYS2TP
Partner LU: MVSLU01
```

### 6.1.3.3 Modifying SYS1.PARMLIB for APPC/MVS

The modifications for APPC that are done in SYS1.PARMLIB affect two members. Those members are APPCPMxx and ASCHPxx. The changes should be done as follows:

APPCPMxx - the diagram in Figure 38 on page 50, indicates changes for SYSTEM1 and SYSTEM2.

```
/********************************************************************/
/*                                                                  */
/*   Define a local LU with the name MVSLU01 to the APPC/MVS        */
/*   configuration.                                                 */
/*                                                                  */
/*   Note:                                                          */
/*                                                                  */
/*   1. The VSAM data set specified on the TPDATA keyword           */
/*      must be already defined to the system, otherwise an         */
/*      error message will be issued.                               */
/*                                                                  */
/*   2. If this LU is intended to handle network traffic,           */
/*      then the LU name MVSLU01 must correspond to the             */
/*      ACBNAME operand of an application definition (APPL)         */
/*      statement in the SYS1.VTAMLST system library and            */
/*      the ACBNAME & APPL name must match.                         */
/*                                                                  */
/********************************************************************/
  LUADD ACBNAME(MVSLU01)     /* Add local LU MVSLU01 to the         */
                             /* APPC/MVS configuration              */
        SCHED(ASCH)          /* Specify that the APPC/MVS           */
                             /* transaction scheduler is associated */
                             /* with this LU name                   */
        BASE                 /* Designate this LU as the base LU    */
        TPDATA(SYS1.APPCTP)  /*                                     */
                             /* RISC.APPCTP is the permanent        */
                             /* repository for the TP profiles      */
                             /* for this LU                         */
        TPLEVEL(GROUP)       /*    TP profiles associated with      */
                             /*    a group of users                 */
  LUADD ACBNAME(MVSLU02)     /* Definition for MVSLU02              */
  TPDATA(SYS1.APPCTP) TPLEVEL(GROUP)  /* MVSLU02 is not the base LU */
/********************************************************************/
/*                                                                  */
/*   Specify that Side information is to be kept in VSAM            */
/*   data set SYS1.APPCSI.                                          */
/*                                                                  */
/*   Note:                                                          */
/*                                                                  */
/*   1. If the SIDEINFO statement is specified with no operands     */
/*      then the default will be:                                   */
/*      o DATASET(SYS1.APPCSI)                                      */
/*                                                                  */
/*   2. The VSAM data set specified on the DATASET keyword          */
/*      must be already defined to the system, otherwise an         */
/*      error message will be issued.                               */
/*                                                                  */
/********************************************************************/
  SIDEINFO DATASET(SYS1.APPCSI)              /*  VSAM data set      */
                                 /* SYS1.APPCSI is the permanent    */
                                 /* repository for the side         */
                                 /* information                     */
```

*Figure 38. APPCPMxx - SYS1.PARMLIB Member for APPC*

**Note:** If the member is already in use, the SIDEINFO can be excluded. This sample is also provided in the BLXAPPCP member of the SBLMSAMP library.

The example of adding CLASS(SHR) is shown in Figure 39 on page 51. The class must be the class specified in the TP profile. It is recommended that this class be used only for this TP profile.

```
/*******************************************************************/
/*                                                                 */
/*   Sample modification to ASCHPMxx member of SYS1.PARMLIB        */
/*   for APPC/MVS Transaction Scheduler                            */
/*   The class has the following characteristics:                 */
/*         o Class name - SHR                                      */
/*         o Maximum number of transaction initiators allowed - 10 */
/*         o Minimum number of transaction initiators to be        */
/*           brought up - 2                                        */
/*         o Response time goal - 0.02 seconds                     */
/*         o Maximum size of the job log for TPs - 500 messages    */
/*                                                                 */
/*******************************************************************/
  CLASSADD CLASSNAME(SHR)    /* Specify the name of the class to be */
                             /* added                              */
           MAX(10)           /* Specify that the maximum number    */
                             /* of transaction initiators allowed  */
                             /* for this class is 10               */
           MIN(2)            /* Specify that the minimum number    */
                             /* of transaction initiators to be    */
                             /* brought up for this class is 2     */
           RESPGOAL(.02)     /* Specify that the response time     */
                             /* goal for transaction programs      */
                             /* executing within this class is 0.02 */
                             /* seconds                            */
           MSGLIMIT(500)     /* Specify that the maximum size of   */
                             /* the job logs for TPs is 500        */
                             /* messages                           */
```

*Figure 39. ASCHPMxx - SYS1.PARMLIB Member for ASCH*

The sample member for ASCH definitions is in the BLXASCHP member of the SBLMSAMP library.

## 6.1.4 VTAM Considerations

Some VTAM modifications must be done for Multisystem Database Access to be implemented successfully. The modification affect the LU definitions, that is, MVSLU01 and MVSLU02 that have been used. As previously stated, the value in the LUADD specified in Figure 38 on page 50 must match the value in the ACBNAME parameter in the VTAM major node. The sample definitions in VTAM for MVSLU01 and MVSLU02 are shown in Figure 40.

```
APPCAPP  VBUILD TYPE=APPL
MVSLU01  APPL  ACBNAME=MVSLU01,APPC=YES,AUTOSES=5,DDRAINL=NALLOW,      X
               DMINWNL=3,DMINWNR=6,DRESPL=NALLOW,DSESLIM=9,EAS=509,    X
               MODETAB=APPCMODE,SECACPT=CONV,VPACING=2,VERIFY=NONE,    X
               SRBEXIT=YES,DLOGMOD=#INTER
MVSLU02  APPL  ACBNAME=MVSLU02,APPC=YES,AUTOSES=5,DDRAINL=NALLOW,      X
               DMINWNL=3,DMINWNR=6,DRESPL=NALLOW,DSESLIM=9,EAS=509,    X
               MODETAB=APPCMODE,SECACPT=CONV,VPACING=2,VERIFY=NONE,    X
               SRBEXIT=YES,DLOGMOD=#INTER
```

*Figure 40. APPCAPP - Sample VTAM Application in SYS1.VTAMLST*

### 6.1.5 Information/Management and APPC Startup

After all the changes above have been set up and completed, the following actions need to be done to activate them:

1. The VTAM major node defined above must be activated with the command:

   ```
   V NET,ACT,ID=APPCAPP
   ```

2. APPC and ASCH must be restarted with the following commands:

   ```
    S APPC,SUB=MSTR,APPC=xx
    S ASCH,SUB=MSTR,ASCH=xx
   ```

   The xx suffix must be modified to match members APPCPMxx and ASCHPMxx of SYS1.PARMLIB as shown in Figure 38 on page 50 and Figure 39 on page 51.

3. Both BLX-SP procedures need to be started. If there is a procedure using SSN BLX1 or BLX2, it must be stopped and then BLX1SH and BLX2SH procedures could be activated.

When updating a record from a session parameter member that uses the BLX1SH service provider address space, the following message will appear on the output log for the BLX2SH service provider:

```
BLX20135I ASID 0033 TCB 008E1D88 USERID BIAS001 connected to the BLX-SP
```

This indicates that communication was established. The message will appear once, for the initial transaction. The message will not appear for the next transaction even though the user ID is different. The response time of the first transaction will be longer than the response times for the subsequent transactions because of the initialization that takes place during the first transaction.

If you experience problems, review your APPC and ASCH setup. Also refer to the APPC documentation. The following commands are available in APPC to check the communication:

- D APPC,TP,ALL - displays information about local and partner TPs and the number of inbound and outbound transactions.
- D APPC,LU,ALL - shows information about each active, outbound, pending and terminating LU.
- D ASCH,ALL - shows information about scheduling classes, active and queued TPs, and the default APPC/MVS scheduling information.

## 6.2 Multisystem Database Access in a Multisystem Environment

As previously mentioned, the concept of Multisystem Database Access is the same in both single and multisystem environments. There is, however, an additional consideration, Global Resource Serialization (GRS), in the latter situation. This topic will cover Global Resource Serialization (GRS), and also refer to diagrams in 6.1, “Multisystem Database Access in a Single System” on page 37. Only the differences in the setup of these components will be highlighted in this section.

Multisystem Database Access was designed for this environment to enable users to share Information/Management databases from one or more MVS/ESA systems.

The diagram in Figure 41 shows Multisystem Database Access in a multisystem environment.

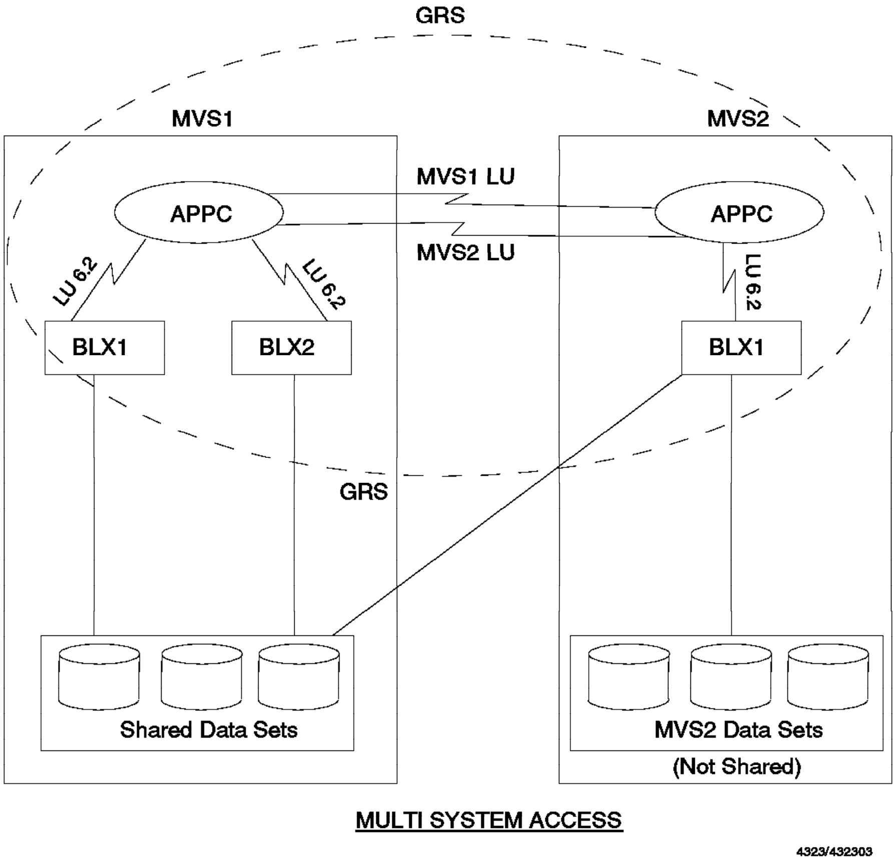


*Figure 41. Multisystem Database Access on Two MVS/ESA Systems*

## 6.2.1 Global Resource Serialization (GRS)

Global Resource Serialization (GRS) is a software product that ensures data integrity amongst multiple systems accessing the same data sets. The complex consists of one or more system connected by communication links. Please refer to manual *MVS/ESA Planning: Global Resource Serialization (GRS)* for more information on tailoring and implementing GRS.

GRS parameters are specified in members IEASYSxx, GRSCNFxx and GRSRNLxx of SYS1.PARMLIB. The following items are the parameters for each member that requires specification:

1. IEASYSxx

   - COUPLE=xx - This parameter is necessary if XCF services are being used in the complex. This describes the sysplex environment.
   - GRS=xxxxxxx - indicates the GRS flag.
   - GRSCNF=xx - the configuration member for GRS.

- GRSRNL=xx - name of the resource name list.
- SYSNAME=xxxx - identifies each system in the complex.

The example of IEASYSxx specification is shown in Figure 42.

```
/********************************************************************/
GRS=TRYJOIN,        PREFERRED
GRSCNF=00,          CONFIGURATION MEMBER LIST
GRSRNL=00,          RESOURCE NAME LIST MEMBER
SYSNAME=SYSTEM1     SYSTEM NAME
```

*Figure 42. IEASYSxx - Sample Entry for GRS in SYS1.PARMLIB*

2. GRSCNFxx

   This member is used to define the global resource serialization complex. Sample specifications are shown in Figure 43.

```
/********************************************************************/
GRSDEF  MATCHSYS(SYSTEM1)       /* MATCHES SYSNAME TO SID.          */
        RESMIL(0)               /* CPU WILL HANDLE MINIMUM          */
                                /* RESONANCE TIME                   */
        CTC(###)                /* CHANNEL ADDRESS THAT LINKS       */
        CTC(###)                /* ON SYSTEM TO ANOTHER             */
        CTC(###)                /* ### IS EQUAL TO CHANNEL ADDRESS  */
        RESTART(YES)            /* AUTO REBUILD CAN DISRUPT RING    */
        REJOIN(YES)             /* REJOIN RING AFTER SYSTEM STOP    */
        TOLINT(20)              /* TOLERATE INTERVAL IS 20 SECONDS  */
        ACCELSYS(2)             /* RING ACCELERATION DEFINITION     */
        CTRACE(CTIRGSKD)        /* COMPONENT TRACE OPTIONS          */
        RESTART(YES)            /* AUTO REBUILD CAN DISRUPT RING    */
```

*Figure 43. GRSCNFxx - Sample Entry in SYS1.PARMLIB*

3. GRSRNLxx

   This member consists of three resource name lists (RNLs). The system uses thess resource name lists to determine how to treat a resource defined in the RNLs. Figure 44 on page 55 shows a sample of this member in SYS1.PARMLIB.

```
/*********************************************************************/
/*  RNLDEF STATEMENT SPECIFYING LINKLIB(YES). IF THIS STATEMENT      */
/*  EXISTS, RNLS WILL BE LOADED FROM THE ISGGRNLO MEMBER OF          */
/*  SYS1.LINKLIB.  IN ORDER TO HAVE RNLS LOADED FROM THE GRSRNL00    */
/*  MEMBER OF SYS1.PARMLIB REMOVE THIS STATEMENT.                    */
/*********************************************************************/

RNLDEF LINKLIB(YES)

/*********************************************************************/
/*  SYSTEMS EXCLUSION RESOURCE NAME LIST - RNLDEF STATEMENTS         */
/*********************************************************************/
RNLDEF RNL(EXCL) TYPE(SPECIFIC)
QNAME(SYSDSN)
RNAME(SYS1.UADS)

/*********************************************************************/
/*  SYSTEM INCLUSION RESOURCE NAME LIST - RNLDEF STATEMENTS          */
/*********************************************************************/
RNLDEF RNL(INCL) TYPE(GENERIC)
QNAME(SYSDSN)
```

*Figure 44. GRSRNLxx - Sample Entry in SYS1.PARMLIB*

The above example can be found in member GRSRNL00 of SYS1.PARMLIB.

### 6.2.2 Information/Management Setup

Customizing Information/Management for Multisystem Database Access involves setting up the environment for MVS1 and MVS2 as shown in Figure 41 on page 53. The following steps need to be done:

1. Set up one started task in SYS1.PROCLIB with the sample JCL shown in Figure 45 on each system. The procedure for BLX2SH as shown in Figure 26 on page 40 is not needed.

```
//BLX1SH   PROC PRM=SH
//BLXSPCAS EXEC PGM=BLXSSP00,REGION=6M,TIME=1440,PARM=&PRM
//STEPLIB  DD   DSN=BLM.V6R1M0.SBLMMOD1       APF AUTHORISED
//BLXPRM   DD   DISP=SHR,BLM.V6R1M0.SBLMSAMP  INFO BLX-SP PARMS
```

*Figure 45. PROCLIB Changes on SYSTEM1 and SYSTEM2*

   There are no additional considerations for BLXSSINM and for the JCL shown in Figure 27 on page 41.

2. Changes to SYS1.PARMLIB

   - Include BLXCBIAS in the program properties table, SCHEDxx. The program must be defined as non-swappable as showh in Figure 46 on page 56.

```
/******************************************************************/
/*                                                                */
/*          PPT FOR INFO V6 MULTISYSTEM DATABASE ACCESS           */
/*                                                                */
/******************************************************************/
  PPT PGMNAME(BLXCBIAS)      /* PROGRAM NAME INFO V6              */
      NOSWAP                 /* NON-SWAPPABLE                     */
```

*Figure 46. SCHEDxx Member of SYS1.PARMLIB*

3. Allocate the Information/Management shared datasets. All data sets that will be shared must be defined with SHROPTIONS (3 3), except for the log data set, which is defined with SHROPTIONS (4 4). These allocations are made so that multiple reads and writes can happen simultaneously.

4. Set up your BLXVDEFx for your local shared resources and non-shared resources. This should include your shared data sets. The data sets should be specified with SHARE=YES as indicated in Figure 29 on page 42.

5. Set up your BLGSESxx member with the Shared data sets.

   Both the BLXVDEF and BLGSESxx must be assembled and link-edited into a load library accessible through a STEPLIB or Link Linklist concatenation.

6. Coding the sample parameter member for the BLX-SP address spaces is the next step in customizing Information/Management. The source shown in Figure 30 on page 43 is the sample BLX1SH member that can be found in the BLM.V6R1M0.SBLMSAMP data set.

   In this environment SYSTEM1 must be the local system and SYSTEM2 the remote system.

### 6.2.3 Advanced Program-to-Program Communication (APPC) Setup

Refer to 6.1.3, "Advanced Program-to-Program Communication (APPC) Setup" on page 44 for a discussion of APPC. The following differences apply:

1. The Transaction Program Profile

   Run the sample JCL in Figure 31 on page 45, since it is on both MVS1 and MVS2 systems.

2. The Side Information Table

   Modify the side information SYSIN card, as shown in Figure 33 on page 47, for both MVS systems as follows:

   ```
   DESTNAME(SYSTEM2)
   TPNAME(SYS2TP)
   PARTNER_LU(MVSLU01)
   ```

3. Modifying SYS1.PARMLIB for APPC/MVS

   The modifications for APPC that are done in SYS1.PARMLIB affect two members, APPCPMxx and ASCHPxx. The following changes should be done:

   APPCPMxx - the diagram in Figure 38 on page 50, indicates changes in MVS1. The LU names must be unique in both MVS/ESA systems. The LU that is not relevant in the figure should be discarded.

   The example of adding CLASS(SHR) as shown in Figure 39 on page 51 could be used for both systems.

### 6.2.4 VTAM Considerations

The discussion in 6.1.4, “VTAM Considerations” on page 51 is applicable. Note, however, that the LU names must be unique for both systems.

### 6.2.5 Information/Management and APPC Startup

Please refer to 6.1.5, “Information/Management and APPC Startup” on page 52 for this topic. The actions should be performed on both systems. However, only considerations made when referring to BLX1 should be used for each MVS/ESA system.

# Chapter 7. Setting Up the CICS/ESA Interface

This section describes how to implement the HLAPI for the CICS environment. The diagram in Figure 47 shows how work passes from CICS, via APPC to the Remote Environment Server (RES). From there, the HLAPI provides the interface to Information/Management.

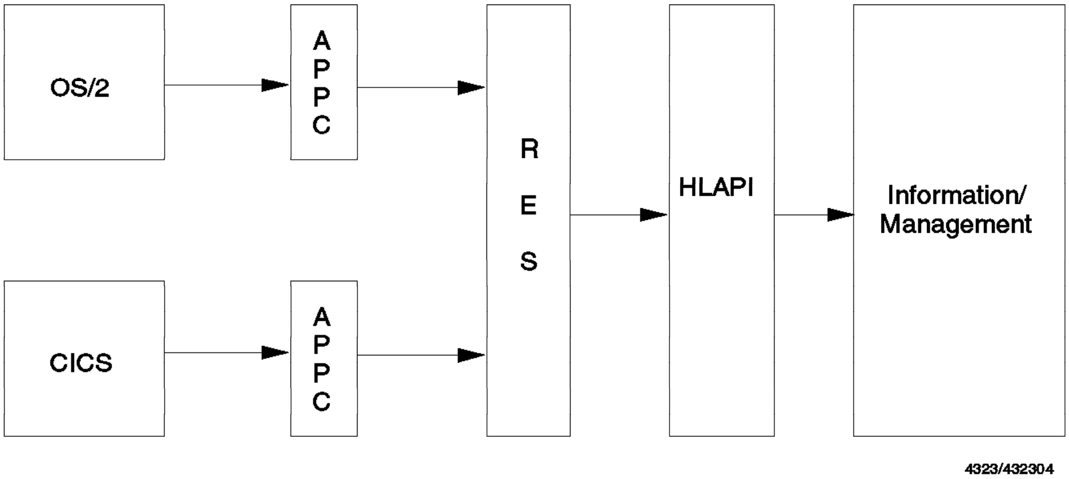


*Figure 47. Overview of the Client Application Programming Interface (CAPI)*

## 7.1 Link-editing the RES and the HLAPI CICS

The Information/Management Version 6 Release 1 Program Directory has two optional steps:

1. Link-edit the RES module

   This step is a prerequisite task for users wishing to implement either HLAPI/2 or the CICS HLAPI interface. (Refer to 5.1, “Remote Environment Server (RES) Module” on page 25 for details.)

2. Link-edit the HLAPI CICS module and sample application code.

   This step is another prerequisite task for people wishing to implement HLAPI CICS. The sample installation JCL provides a job BLMYKLNK. This performs the following:

   - Link-edit a load module, BLMYSCSC, outside of SMP/E control, which integrates your CICS DFHEAI stub as the first CSECT of the module BLMYKINF.

   - Link-edit the CICS programs that form the CICS API sample application.

This client, also referred to as HLAPI CICS, utilizes standard CICS function to invoke the Information/Management Remote Environment Server (RES) to handle API calls. The following tasks will be described here:

- Setting up Advanced Program-to-Program Communication (APPC) link between CICS/ESA and APPC/MVS

- Setting up CICS transaction definitions
- Setting up CICS program definitions

A complete description of the API transactions can be found in the *Application Programming Interface (API) Guide and Reference,* SC34-4483. Please note that not all these functions are available in the samples described in 7.5, "Installing the Definitions to Support the Sample Code" on page 71.

The HLAPI CICS interface provides the following functions:

- Begin Information/Management (API transaction HL01)
- End Information/Management (API transaction HL02)
- Obtain external record ID (API transaction HL03)
- Check Out Record (API transaction HL04)
- Check In Record (API transaction HL05)
- Retrieve Record (API transaction HL06)
- Create Record (API transaction HL08)
- Update Record (API transaction HL09)
- Record Inquiry (API transaction HL11)
- Add Record Relations (API transaction HL12)
- Delete Record (API transaction HL13)

## 7.2 Establishing an LU 6.2 Link between CICS/ESA and APPC/MVS

There are two sides to the link between CICS/ESA and APPC/MVS:

- APPC/MVS definitions
- CICS/ESA definitions

You must define the resources on both sides. This section will detail how to define the resources to APPC. 7.3, "Defining Resources to CICS/ESA" on page 62 will describe how to define the CICS resources.

An APPC conversation sends data back and forth between two programs running in two host systems. For the purpose of our discussion, one program can be CICS/ESA and the other can be Information/Management, or both can be Information/Management. In the latter case, see Chapter 6, "Multisystem Database Access" on page 37.

APPC uses two key resources to control the communication:

**TP** The TP name is the name you are assigning to a transaction program which will handle the exchange of data. There are two types of TP:

- The local TP. This TP initiates the activity. In this section, CICS/ESA is the local TP.
- The partner TP. This is the server TP defined to APPC/MVS.

**Symbolic Destination** This is an entry in the side information table, and essentially it is the means by which we tell APPC who it is talking to.

There are two ways in which we can define the TP name and side information to APPC/MVS: through ISPF dialogs or by submitting batch jobs.

## 7.2.1 ISPF Panels

This method was described in 5.2, “Preparing APPC/MVS Definitions” on page 25. The same procedure can be adapted, using the appropriate names, for defining connections with CICS.

## 7.2.2 Batch Jobs

There are two jobs to run, as follows:

1. ADDTP

   This job creates the TP definition in the APPCTP file. You will also see that this creates some JCL, which is the Remote Environment Server (RES) JCL. This will be used by APPC the first time an attempt is made to start an HL01 API transaction from a CICS program, to run the Remote Environment Server (RES) address space. There will be one of these address spaces running for each concurrent API user - that is, a user in CICS who has run a program that has started an HL01, but has not yet run a HL02. The HL02 API transaction will cause the Remote Environment Server (RES) address space to be terminated.

```
//jobname  JOB you-job-card
//*
//*************************************************************
//* THE FOLLOWING DEFAULTS ARE ASSUMED:
//*        I) TYPRUN=RUN
//*       II) TPSCHED_EXIT(ASCH)
//*************************************************************
//STEP1    EXEC PGM=ATBSDFMU
//SYSSDLIB  DD DSN=SA28.APPCTP,DISP=SHR
//SYSSDOUT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DATA,DLM=XX
TPADD
   TPNAME(INFOCTP1)
   ACTIVE(YES)
   TPSCHED_DELIMITER(##)
      CLASS(FAST)
      TPSCHED_TYPE(STANDARD)
      JCL_DELIMITER(END_OF_JCL)
//BLMYSCSC JOB  MSGLEVEL=(1,1),MSGCLASS=A
//YSCSC    EXEC PGM=BLMYSCSC
//STEPLIB   DD   DISP=SHR,DSN=BLM610.BASE.SBLMMOD1
//SYSPRINT  DD SYSOUT=*
//SYSTSIN   DD DUMMY
//SYSTSPRT  DD SYSOUT=*
//SYSIN     DD DUMMY
END_OF_JCL
   KEEP_MESSAGE_LOG(ERROR)
##
XX
/*
//
```

*Figure 48. Job to Define TP Name for CICS*

2. ADDSI

When APPC needs to send data back to CICS, it uses this information to direct the data. The PARTNER_LU is the VTAM APPLID of the CICS region that is running HLAPI CICS.

```
//jobname  JOB you-job-card
//*
//STEP1    EXEC PGM=ATBSDFMU
//SYSSDLIB  DD DSN=SA28.APPCSI,DISP=SHR
//SYSSDOUT  DD SYSOUT=*
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
 SIADD
   DESTNAME(INFOSIP1)
   TPNAME(INFOCTP1)
   PARTNER_LU(RAKAC001)
/*
//
```

*Figure 49. Job to Define Side Information for CICS*

## 7.3 Defining Resources to CICS/ESA

There are two sets of definitions: one is mandatory to be able to implement the CICS/ESA feature, and the second set tells CICS/ESA about the sample programs that have been provided with Information/Management Version 6 Release 1.

## 7.4 Installing the Mandatory Definitions

The following places in CICS/ESA require customization:

- System Initialization Table (SIT)

  This is the CICS table where you code parameters that are used at CICS startup time to influence the CICS environment.

- Destination Control Table (DCT)

  This is the CICS table where you define the names of queues, which can be considered scratch pads. HLAPI CICS uses one of these to write its log information.

- Shutdown Program Load Table (PLT)

  There are in fact two PLT tables. These tell CICS to execute certain programs as part of its startup or shutdown procedures.

- Program Control Table (PCT)

  This is the CICS table which defines the names of transactions which are going to run in the CICS system.

- Program Processing Table (PPT)

  This is the CICS table which defines the names of programs which are going to run in the CICS system.

- Terminal Control Table (TCT)

This is the CICS table which defines terminals and ISC links which are going to be used in the CICS system.

- CICS/ESA System definition data set (DFHCSD)

  This is a VSAM file which contains the PCT, PPT, and TCT definitions for the CICS system.

- CICS/ESA startup JCL

### 7.4.1 Customizing the SIT

There are three ways to achieve this as shown below. However, the recommended way is to use overrides in a SYSIN data set. CICS SIT parameters are described in the *CICS/ESA System Definition Guide*.

- Using Overrides

  This is done by using a SYSIN data set in the startup JCL for CICS, in the DFHSIP step. Code the following parameters in this data set, if they are not already included.

  – ISC=YES
  – PLTSD=IM
  – DCT=IM
  – DFLTUSER=user

- Modifying the startup procedure parameter

  Make the same changes described above, but code them as a string on the PARM field of the EXEC PGM=DFHSIP card of your CICS startup JCL.

- Changing the source

  Change the source of DFHSITxx for your CICS/ESA system to include the same parameter names as above; then assemble and link-edit the table with JCL similar to that shown in Figure 50. Please note that with CICS/ESA Version 3.3, if a SIT is assembled, it must be assembled into an authorized library. It is for this reason that this approach is not recommended.

```
//jobname      JOB your-job-card
//*
//ASMTAB   EXEC PROC=DFHAUPLE,NAME=SDFHAUTH
//*
//ASSSEM.SYSUT1 DD DSN=your.table.source(DFHSITxx),DISP=SHR
```

*Figure 50. Job to Define Side Information CICS*

### 7.4.2 Customizing the DCT

Update the source of DFHDCTxx for your CICS/ESA system, with the changes shown in Figure 52 on page 64. Then assemble and link-edit this with JCL similar to Figure 52 on page 64, but this time do not assemble it into your authorized library. The figure shows NAME=SDFHLOAD, which is the third qualifier on the SYSLMOD data set. It is likely that you will assemble this into your installation-defined CICS table library, which is concatenated ahead of SDFHLOAD in the CICS JCL.

```
BLMLOG   DFHDCT TYPE=SDSCI,        INFO/MAN HLAPI CICS LOG             X
               BLKSIZE=136,                                            X
               BUFNO=1,                                                X
               DSCNAME=BLMLOG,                                         X
               RECFORM=VARBLK,                                         X
               RECSIZE=132,                                            X
               TYPEFLE=OUTPUT
*
BLML     DFHDCT TYPE=EXTRA,        INFO/MAN DCT                        X
               DESTID=BLML,                                            X
               DSCNAME=BLMLOG
*
```

*Figure 51. Information/Management CICS DCT Definitions*

```
//jobname      JOB your-job-card
//*
//ASMTAB   EXEC PROC=DFHAUPLE,NAME=SDFHLOAD
//*
//ASSSEM.SYSUT1 DD DSN=your.table.source(DFHDCTxx),DISP=SHR
```

*Figure 52. Information/Management CICS DCT Assembly JCL*

### 7.4.3 Customizing the PLT

Update the source of your shutdown PLT, DFHPLTxx, where xx is the installation-defined suffix. Figure 53 shows a skeleton PLT. Please note that there two types of shutdown program: first pass and second pass. BLMYKTRM is a first pass program, which means it should go before the PROGRAM=DFHDELIM card. Assemble and link-edit this with JCL similar to Figure 54 on page 65. Again, this should not be written to your authorized library.

```
         DFHPLT TYPE=INITIAL,                                          X
               SUFFIX=SD
PLTSD    DFHPLT TYPE=ENTRY,PROGRAM=........ (your program)
         DFHPLT TYPE=ENTRY,PROGRAM=BLMYKTRM
         DFHPLT TYPE=ENTRY,PROGRAM=........ (your program)
         DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
         DFHPLT TYPE=ENTRY,PROGRAM=........ (your program)
         DFHPLT TYPE=FINAL
         END   ,
```

*Figure 53. Information/Management CICS PLT Definitions*

```
//jobname      JOB your-job-card
//*
//ASMTAB   EXEC PROC=DFHAUPLE,NAME=SDFHLOAD
//*
//ASSSEM.SYSUT1 DD DSN=your.table.source(DFHPLTxx),DISP=SHR
```

*Figure 54. Information/Management CICS PLT Assembly JCL*

## 7.4.4 Customizing the Startup JCL

Only one change needs to be made to the CICS startup JCL. Add an entry for DD nane BLMLOG, which is used by HLAPI CICS to log activity that has taken place. You can do this in either of the following ways:

1. `//BLMLOG DD DSN=your.hlapi.cics.log,DISP=SHR`
2. `//BLMLOG DD SYSOUT=<class>`

## 7.4.5 Customizing the PCT, PPT, and TCT

These three tables are defined in the CICS System Definition file (CSD), and entries can be defined in them dynamically, either through the interactive CEDA transaction, or through a batch job executing program DFHCSDUP.

### 7.4.5.1 Using CEDA to Update the CICS PCT

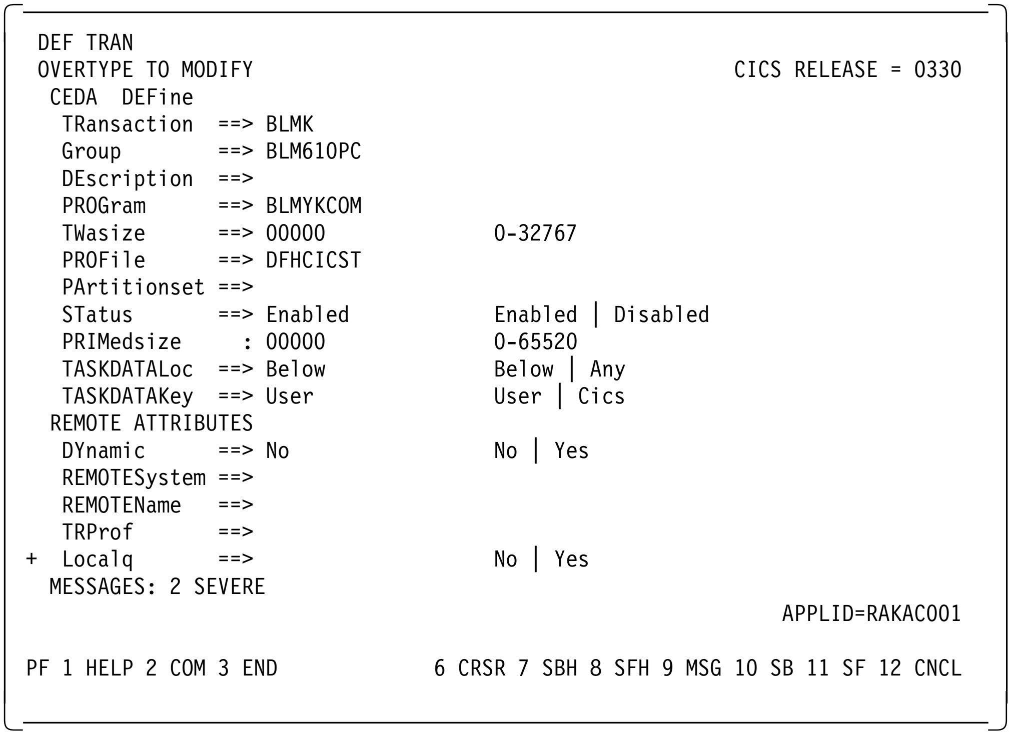
The instructions below show how to define a transaction to CICS, using the online CEDA transaction. Two transactions need to be defined: BLMK, which calls program BLMYKCOM, and BLMT, which calls program BLMYKTRM.

1. Sign on to CICS.
2. Type `CEDA DEF TRAN` and press Enter. You will be presented with the screen in Figure 55 on page 66.
3. Enter the following information.
   - The transaction code - BLMK
   - The name of the group (you can use any name you wish, but be consistent. We suggest BLM610PC)
   - The name of the program that this transaction will execute - BLMYKCOM
   - To allow the timeout value to purge a long waiting task, code `SPURGE=YES`
   - To allow the CICS transaction to run above the 16 MB line, code `TASKDATALOC=ANY`
4. Press Enter to tell CEDA to accept these values. CEDA will give you a message `Define Successful` in the bottom left corner of the screen.
5. Press Enter now and your screen will have `DEF TRAN` in the top left corner, and still be under CEDA′s control.
6. Press Enter now, and you can define a second transaction. Enter the following details for this transaction.
   - The transaction code - BLMT
   - The name of the group - again we suggest BLM610PC
   - The name of the program that this transaction will execute - BLMYKTRM
   - To allow the timeout value to purge a long waiting task, code `SPURGE=YES`

- To allow the CICS transaction to run above the 16 MB line, code `TASKDATALOC=ANY`

7. If you wish to define another transaction, press Enter at this point and you will be able to repeat the above steps. If not, press PF3 followed by the Clear key to return to native CICS.

   If you have now successfully defined two transactions, proceed to define the programs.



```
 DEF TRAN
 OVERTYPE TO MODIFY                                        CICS RELEASE = 0330
  CEDA  DEFine
   TRansaction  ==> BLMK
   Group        ==> BLM610PC
   DEscription  ==>
   PROGram      ==> BLMYKCOM
   TWasize      ==> 00000             0-32767
   PROFile      ==> DFHCICST
   PArtitionset ==>
   STatus       ==> Enabled           Enabled | Disabled
   PRIMedsize     : 00000             0-65520
   TASKDATALoc  ==> Below             Below | Any
   TASKDATAKey  ==> User              User | Cics
  REMOTE ATTRIBUTES
   DYnamic      ==> No                No | Yes
   REMOTESystem ==>
   REMOTEName   ==>
   TRProf       ==>
+  Localq       ==>                   No | Yes
  MESSAGES: 2 SEVERE
                                                               APPLID=RAKAC001

PF 1 HELP 2 COM 3 END           6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

*Figure 55. CICS/ESA - The CEDA Define Transaction Screen*

### 7.4.5.2 Using CEDA to Update the CICS PPT

The instructions below give advice on how to define a program to CICS using the online CEDA transaction. Three program need to be defined: BLMYKINF, BLMYKCOM and BLMYKTRM.

1. Sign on to CICS, if you are not already signed on.

2. Type `CEDA DEF PROG` and press Enter. You will be presented with the screen in Figure 56 on page 67.

3. On the screen in Figure 56 on page 67, you will need to enter four pieces of information, and the rest can take the default values.

   - The program name - BLMYKCOM
   - The name of the group (you can use any name you wish, but be consistent. We suggest BLM610PP)
   - The programming language - ASSEMBLER
   - Datalocation - Set this to `any`

4. Press Enter to tell CEDA to accept these values. CEDA will give you a message `Define Successful` in the bottom left corner of the screen.

5. Press Enter now and your screen will have DEF PROG in the top left corner, and still be under CEDA′s control.

   Press Enter. Then you can define a second program with the following values:

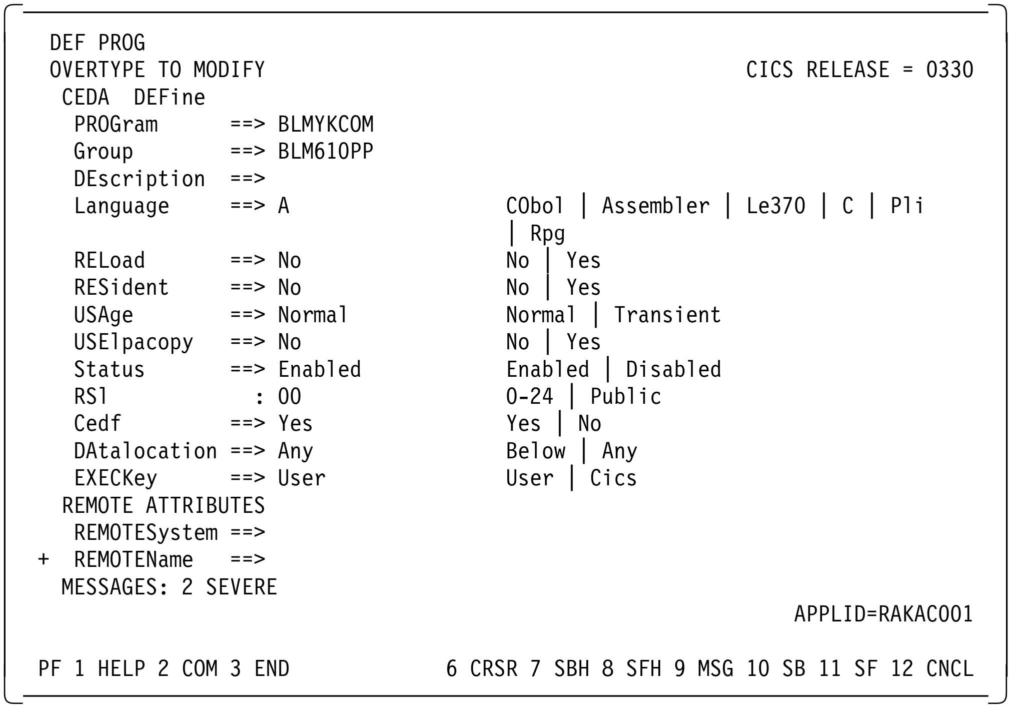
   - The program name - BLMYKINF
   - The name of the group (you can use any name you wish, but be consistent. We suggest BLM610PP)
   - The programming language - ASSEMBLER
   - Datalocation - Set this to any

   Press Enter. Then you can define a third program, with the following values:

   - The program name - BLMYKTRM
   - The name of the group (you can use any name you wish, but be consistent. We suggest BLM610PP)
   - The programming language - ASSEMBLER
   - Datalocation - Set this to any

6. If you wish to define another program, press Enter at this point and you will be able to repeat the above steps. If not, press PF3 followed by the Clear key to return to native CICS.

   If you have now successfully defined all the required programs, then proceed to define the connection.



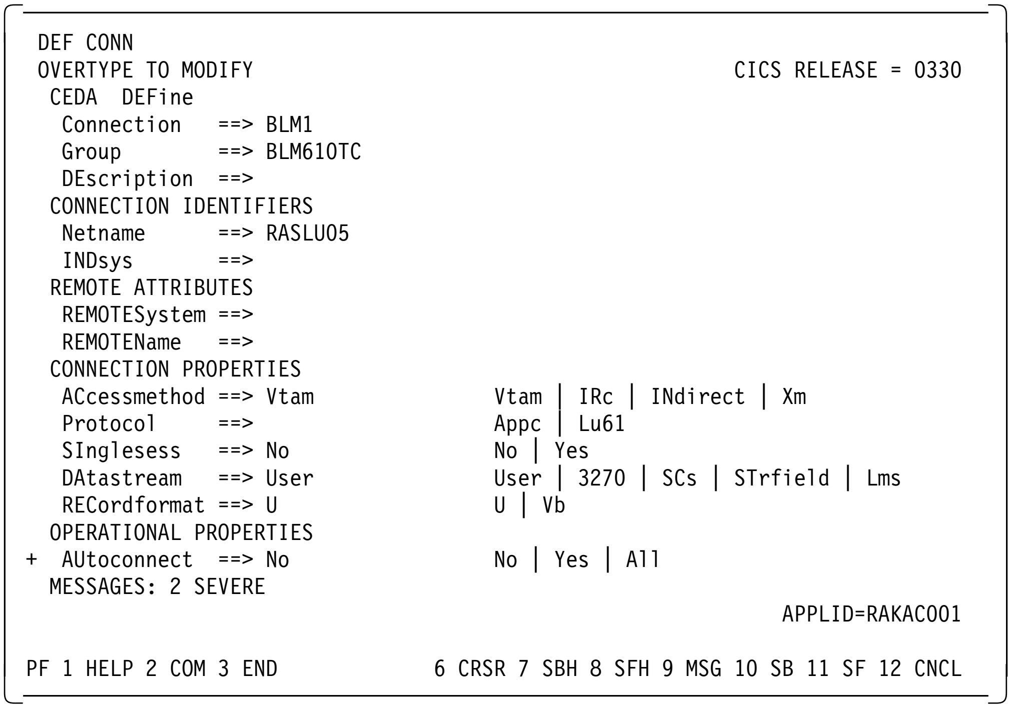
```
 DEF PROG
 OVERTYPE TO MODIFY                                        CICS RELEASE = 0330
  CEDA  DEFine
   PROGram      ==> BLMYKCOM
   Group        ==> BLM610PP
   DEscription  ==>
   Language     ==> A                  CObol | Assembler | Le370 | C | Pli
                                       | Rpg
   RELoad       ==> No                 No | Yes
   RESident     ==> No                 No | Yes
   USAge        ==> Normal             Normal | Transient
   USElpacopy   ==> No                 No | Yes
   Status       ==> Enabled            Enabled | Disabled
   RSl            : 00                 0-24 | Public
   Cedf         ==> Yes                Yes | No
   DAtalocation ==> Any                Below | Any
   EXECKey      ==> User               User | Cics
  REMOTE ATTRIBUTES
   REMOTESystem ==>
+  REMOTEName   ==>
  MESSAGES: 2 SEVERE
                                                               APPLID=RAKAC001

PF 1 HELP 2 COM 3 END             6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

Figure 56. CICS/ESA - The CEDA Define Program Screen

### 7.4.5.3 Using CEDA to Update the CICS TCT

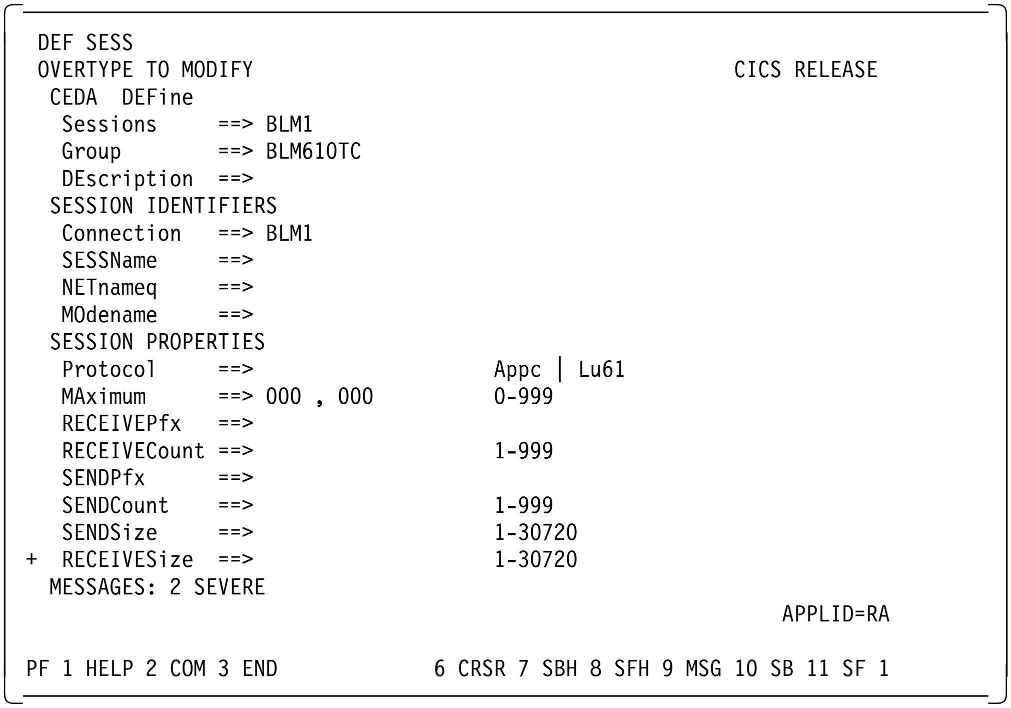
The instructions below give advice on how to define an APPC connection to CICS, using the online CEDA transaction. Only one link needs to be defined.

1. Sign on to CICS, if you are not already signed on.
2. Invoke the CEDA transaction, by typing `CEDA DEF CONN` and you will be presented with the screen in Figure 57 on page 69.
3. On the screen in Figure 57 on page 69, you will need to enter two pieces of information, and the rest can take the default values.
   - The name of the connection - we suggest BLM1
   - The name of the group (you can use any name you wish, but be consistent. We suggest BLM610TC)
4. Press Enter to tell CEDA to accept these values. CEDA will give you a message `Define Successful` in the bottom left corner of the screen.
5. Press Enter now and your screen will have `DEF CONN` in the top left corner, and still be under CEDA′s control.
6. Overtype this with `DEF SESS`, and you will be presented with the screen in Figure 58 on page 69.
7. On the screen in Figure 58 on page 69, you will need to enter two pieces of information, and the rest can take the default values.
   - The name of the sessions - we suggest BLM1 again
   - The name of the connection - BLM1
8. Press Enter to tell CEDA to accept these values. CEDA will give you a message `Define Successful` in the bottom left corner of the screen.
9. Press Enter now and your screen will have `CEDA DEF SESS` in the top left corner, and still be under CEDA′s control.
10. Overtype this with `DEF PARTNER`, and you will be presented with the screen in Figure 59 on page 70.
11. On the screen in Figure 59 on page 70, you will need to enter the following of information, and the rest can take the default values.
    - The name of the partner - we suggest BLM1PART - this will be referred to in the sample panels
    - The name of the group - BLM610TC
    - The netname of the APPC/MVS LU
    - The network name on which the APPC/MVS LU resides
    - The TP name - this needs to match the value in Figure 48 on page 61
12. Press Enter to tell CEDA to accept these values. CEDA will give you a message `Define Successful` in the bottom eft corner of the screen.
13. Press Enter now and your screen will have `DEF PARTNER` in the top left corner, and still be under CEDA′s control.
14. Press PF3 followed by the Clear key to return to native CICS.
15. You have now defined an LU 6.2 link to CICS.



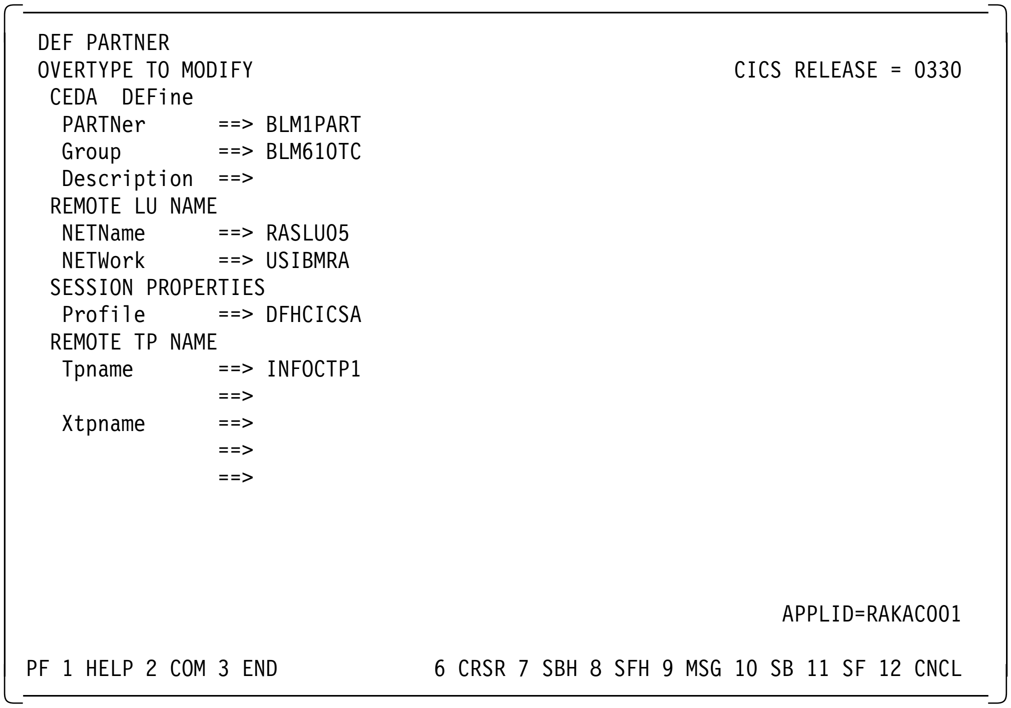
```
 DEF CONN
 OVERTYPE TO MODIFY                                        CICS RELEASE = 0330
  CEDA  DEFine
   Connection   ==> BLM1
   Group        ==> BLM610TC
   DEscription  ==>
  CONNECTION IDENTIFIERS
   Netname      ==> RASLU05
   INDsys       ==>
  REMOTE ATTRIBUTES
   REMOTESystem ==>
   REMOTEName   ==>
  CONNECTION PROPERTIES
   ACcessmethod ==> Vtam              Vtam | IRc | INdirect | Xm
   Protocol     ==>                   Appc | Lu61
   SInglesess   ==> No                No | Yes
   DAtastream   ==> User              User | 3270 | SCs | STrfield | Lms
   RECordformat ==> U                 U | Vb
  OPERATIONAL PROPERTIES
+  AUtoconnect  ==> No                No | Yes | All
  MESSAGES: 2 SEVERE
                                                               APPLID=RAKAC001

PF 1 HELP 2 COM 3 END            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

*Figure 57. CICS/ESA - The CEDA Define Connection Screen*



```
 DEF SESS
 OVERTYPE TO MODIFY                                        CICS RELEASE
  CEDA  DEFine
   Sessions     ==> BLM1
   Group        ==> BLM610TC
   DEscription  ==>
  SESSION IDENTIFIERS
   Connection   ==> BLM1
   SESSName     ==>
   NETnameq     ==>
   MOdename     ==>
  SESSION PROPERTIES
   Protocol     ==>                   Appc | Lu61
   MAximum      ==> 000 , 000         0-999
   RECEIVEPfx   ==>
   RECEIVECount ==>                   1-999
   SENDPfx      ==>
   SENDCount    ==>                   1-999
   SENDSize     ==>                   1-30720
+  RECEIVESize  ==>                   1-30720
  MESSAGES: 2 SEVERE
                                                               APPLID=RA

PF 1 HELP 2 COM 3 END            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 1
```

*Figure 58. CICS/ESA - The CEDA Define Connection Screen*



```
 DEF PARTNER
 OVERTYPE TO MODIFY                                       CICS RELEASE = 0330
  CEDA  DEFine
   PARTNer      ==> BLM1PART
   Group        ==> BLM610TC
   Description  ==>
  REMOTE LU NAME
   NETName      ==> RASLU05
   NETWork      ==> USIBMRA
  SESSION PROPERTIES
   Profile      ==> DFHCICSA
  REMOTE TP NAME
   Tpname       ==> INFOCTP1
                ==>
   Xtpname      ==>
                ==>
                ==>



                                                             APPLID=RAKAC001

PF 1 HELP 2 COM 3 END            6 CRSR 7 SBH 8 SFH 9 MSG 10 SB 11 SF 12 CNCL
```

*Figure 59. CICS/ESA - The CEDA Define Partner Screen*

### 7.4.5.4 Using the Batch Utility DFHCSDUP

CICS/ESA supplies an alternative way to update the DFHCSD file, and that is through the batch utility, DFHCSDUP. The diagram in Figure 60 on page 71 shows an example of a batch job containing SYSIN cards to create the same definitions that were created with the online CEDA transaction above. Please refer to *CICS/ESA Resource Definition (Online), SC33-0666* for further details about DFHCSDUP. The following changes should be made before submitting this job:

1. Change the jobcard to suit your installation standards.
2. Change the name of the CICS library in the STEPLIB card to a valid CICS/ESA library for your installation.
3. Change the name of the DFHCSD to the name of the CSD file used by the CICS system that will be supporting HLAPI CICS.
4. Change NETNAME, NETWORK, and TPNAME to match the values that you have already defined in APPC/MVS.
5. Modify the group names for the resources in the SYSIN if they are inconsistent with installation standards for names of CICS resources.
6. Modify the LIST name in the ADD GROUP statement if it is inconsistent with installation standards for names of CICS resources.
7. Verify that the PARTNER name is the same as that in CICS_PARTNER_ID_PDB.

```
//jobcard
//*
//*                                                                     00020000
//* UPDATE THE CSD                                                      00039000
//*                                                                     00040000
//CSDUP    EXEC PGM=DFHCSDUP,REGION=4M                                  00050000
//STEPLIB   DD DISP=SHR,DSN=your.SDFHLOAD                               00060003
//*         DD DISP=SHR,DSN=your.table.lib                              00061002
//DFHCSD    DD DISP=SHR,DSN=your.appl.DFHCSD                            00070002
//SYSPRINT  DD SYSOUT=*                                                 00080000
//SYSUDUMP  DD SYSOUT=*                                                 00090000
//SYSIN     DD *

 DEFINE PROGRAM(BLMYKINF) GROUP(BLM610PP)
   LANGUAGE(ASSEMBLER) DATALOCATION(ANY)
   RES(YES)

 DEFINE PROGRAM(BLMYKCOM) GROUP(BLM610PP)
   LANGUAGE(ASSEMBLER) DATALOCATION(ANY)
   RES(YES)

 DEFINE PROGRAM(BLMYKTRM) GROUP(BLM610PP)
   LANGUAGE(ASSEMBLER) DATALOCATION(ANY)
   RES(YES)


 DEFINE TRANSACTION(BLMK) GROUP(BLM610PC)
   TASKDATALOC(ANY)
   PROGRAM(BLMYKCOM)

 DEFINE TRANSACTION(BLMT) GROUP(BLM610PC)
   TASKDATALOC(ANY)
   PROGRAM(BLMYKTRM)


 DEFINE CONNECTION(BLM1) GROUP(BLM610TC)

 DEFINE SESSION(BLM1) CONNECTION(BLM1) GROUP(BLM610TC)

 DEFINE PARTNER(BLM1PART) GROUP(BLM610TC)
     NETNAME(TBA)
     NETWORK(GBIBMFG)
     TPNAME(TBA)


 ADD GROUP(BLM610PP) LIST(BLM6LIST)
 ADD GROUP(BLM610PC) LIST(BLM6LIST)
 ADD GROUP(BLM610TC) LIST(BLM6LIST)
/*
//jobcard
```

*Figure 60. CICS/ESA - Using DFHCSDUP to Implement CICS/ESA Feature*

## 7.5 Installing the Definitions to Support the Sample Code

The sample code consists of a menu transaction, and transactions that are called from that menu to run the following functions:

- Start API using HL01
- End API using HL02

- Create using HL08
- Delete using HL13
- Retrieve using HL06

Use the instructions in 7.4.5, "Customizing the PCT, PPT, and TCT" on page 65 for using the CEDA transactions to define the following resources to CICS. Alternatively, use a batch job similar to the sample in Figure 61 on page 73, to achieve the same end result.

- Transactions:
  - BLMM
  - BLM1
  - BLM2
  - BLM3
  - BLM6
  - BLM8
  - BLMD
  - BLME
- Programs:
  - BLMYKMNU
  - BLMYKCTL
  - BLMYKCRE
  - BLMYKDEL
  - BLMYKRTV
- Mapset BLMYKMAP

  A Mapset is defined in a similar way to a PROGRAM, except substitute `DEF MAPSET` for `DEF PROG` in the checklist in 7.4.5, "Customizing the PCT, PPT, and TCT" on page 65.

### 7.5.1 How to Use the Sample Code

The sample code is invoked by signing on to CICS, and executing transaction BLMM to bring up the sample menu. From there, follow the instructions on the panels to talk across the link to Information/Management.

For more information on using the samples, see *Jumping out of the Box: New Interfaces and Features of Information/Management Version 6 Release 1*.

```
//jobcard
//*
//* UPDATE THE CSD
//*
//CSDUP    EXEC PGM=DFHCSDUP,REGION=4M
//STEPLIB   DD DISP=SHR,DSN=yourcics.SDFHLOAD
//DFHCSD    DD DISP=SHR,DSN=yourcics.DFHCSD
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
*
 DEFINE PROGRAM(BLMYKMNU) GROUP(BLM610SP)
   LANGUAGE(COBOL) DATALOCATION(ANY)
   RES(YES)
*
 DEFINE PROGRAM(BLMYKCTL) GROUP(BLM610SP)
   LANGUAGE(COBOL) DATALOCATION(ANY)
   RES(YES)
*
 DEFINE PROGRAM(BLMYKRTV) GROUP(BLM610SP)
   LANGUAGE(COBOL) DATALOCATION(ANY)
   RES(YES)
*
 DEFINE PROGRAM(BLMYKCRE) GROUP(BLM610SP)
   LANGUAGE(COBOL) DATALOCATION(ANY)
   RES(YES)
*
 DEFINE PROGRAM(BLMYKDEL) GROUP(BLM610SP)
   LANGUAGE(COBOL) DATALOCATION(ANY)
   RES(YES)
*
 DEFINE MAPSET(BLMMAPS) GROUP(BLM610SP)
   RES(YES)
*
 DEFINE TRANSACTION(BLMM) GROUP(BLM610ST)
   TASKDATALOC(ANY)   PROGRAM(BLMYKMNU)
*
 DEFINE TRANSACTION(BLM1) GROUP(BLM610ST)
   TASKDATALOC(ANY)   PROGRAM(BLMYKCTL)
*
 DEFINE TRANSACTION(BLM2) GROUP(BLM610ST)
   TASKDATALOC(ANY)   PROGRAM(BLMYKCTL)
*
 DEFINE TRANSACTION(BLM6) GROUP(BLM610ST)
   TASKDATALOC(ANY)   PROGRAM(BLMYKRTV)
*
 DEFINE TRANSACTION(BLM8) GROUP(BLM610ST)
   TASKDATALOC(ANY)   PROGRAM(BLMYKCRE)
*
 DEFINE TRANSACTION(BLMD) GROUP(BLM610ST)
   TASKDATALOC(ANY)   PROGRAM(BLMYKDEL)
*
 DEFINE TRANSACTION(BLME) GROUP(BLM610ST)
   TASKDATALOC(ANY)   PROGRAM(BLMYKDEL)
*
*

*
*
 ADD GROUP(BLM610SP) LIST(BLM6LIST)
 ADD GROUP(BLM610ST) LIST(BLM6LIST)
/*
```

*Figure 61. CICS/ESA - Sample DFHCSDUP*

## 7.6 Hints and Tips

If you get a return code 12, reason code 118, then check the following:

- The status of the LU 6.2 link, as CICS sees it. From CICS, issue a `CEMT I CONN` for your link to APPC/MVS. This should be in `INS ACQ status`.
- APPC Definitions.
- Whether you have specified that symbolic destination name as CICS sees it.

If you get a return code 12, reason code 33, then check the following:

- You have *all* Information/Management data sets defined.
- You have a seven-character user ID in your DFLTUSER parameter in your CICS SIT overrides.

The following diagnostics will help solve problems with HLAPI CICS:

1. Turn on LLAPI and HLAPI logging with the control PBDs:

   ```
   APIMSG_OPTION = ′P′
   HLIMSG_OPTION = ′P′
   SPOOL_INTERVAL = ′2′
   ```

2. Add a card for DDNAME APIPRINT to your Remote Environment Server (RES) JCL
3. Add a card for DDNAME HLAPILOG to your Remote Environment Server (RES) JCL

# Chapter 8. Increasing Availability

This section describes how to implement the Automatic Log Save Facility and how to copy updates from a prime Information/Management database to a Hot Standby database. This second database can be on the same MVS machine, or on a remote MVS machine.

Mirroring data improves reliability, availability, and performance for users in the following ways:

- Information/Management production databases can be backed up automatically, on a scheduled time period, specified by the customer.
- The backup database can be used for reports or recovery.
- Administrators can specify an automatic, time-driven offload of the log data set (SDLDS), reducing the risk of running out of space on the SDLDS.

## 8.1 How the Automatic Log Save Works

The diagram in Figure 62 on page 76 gives a simple overview of the logic flow of the batch jobs which implement this feature. The sequence of events is summarized as follows:

1. The first Send job is submitted onto the system. This will cause an internal version of the BLGUT4 utility to offload SDLDS log updates to a sequential file. The SDLDS is mandatory on the sending system.
2. The JCL has condition code checking that asks whether the offload has worked, and if so, it builds and submits the JCL for the Receive job to run against the second database.
3. The Receive job runs an internal version of the BLGUT3 utility to apply the updates from the sequential data set to the second database. The second database must not have an SDLDS.
4. Condition code checking in the Receive job will cause the next Send job to be submitted on to the system,and take you around the loop again.

If the jobs continue to complete successfully, then this loop will continue until one of the following events occurs:

- The BLX1 server is stopped.
- One of the Log Save jobs is cancelled.
- The LOGSAVE record on either database is set to DISABLED.

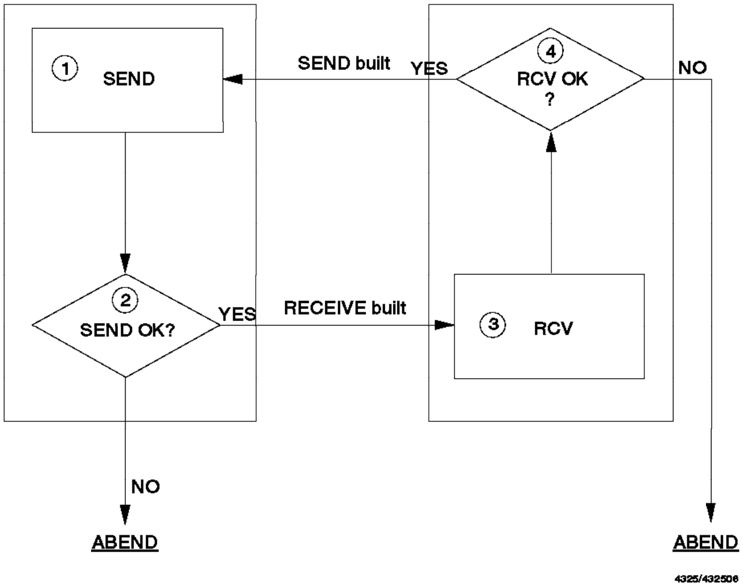


*Figure 62. Automatic Log Save Logic Flow*

## 8.2 Understanding the 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 this Send data set in-stream as part of a job to perform the Automatic Log Save Receive function.

A new type of record is available called the LOGSAVE record. The LOGSAVE record provides a place to store the processing options:

- Send data set characteristics.
- Synchronize (yes or no) the send and receive functions.
- Time interval to offload the SDLDS.

The Automatic Log Save Send function is started by the Send job BLMSASD. The Send job performs the following tasks:

1. Starts TSP BLGDUMP1:
   a. Call user exit BLGUT4WT to

        1) Read the LOGSAVE record.

        2) Verify Send database status.

        3) Verify LOGSAVE record status enable.

        4) Put LOGSAVE record information in the TSCA variable data area.

        5) Wait time as specified in LOGSAVE record, then return to TSP.

    b. Call user exit BLGUT4EX to:

        1) Read TSCA variable data area.

        2) Allocate Send data set.

        3) If synchronization is set on:

            a) Obtain 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 to the sequential Send data set.

        5) If synchronization is set on, store the sequence number in the SDIDS.

2. Sends a notification to a specified user ID.

3. Copies the Send data set to the GDG.

4. Build and route the Receive job BLMSARV.

5. The Send function is complete.

## 8.3 Understanding the Automatic Log Save Receive Function

The Automatic Log Save Receive function receives the send data set from the Automatic Log Save Receive function into the receive data set. It then adds the records from the sequential Receive data set to the Information/Management receive database.

The LOGSAVE record on the Receive system is used to determine the following:

- Receive data set characteristics.
- Synchronize (yes or no) the Send and Receive functions.
- Time interval to process the Receive data set.
- Filtering criteria.

The Automatic Log Save Receive function is initiated by the Automatic Log Save Send function through the Receive job BLMSARV. The Receive job performs the following tasks:

1. Deletes existing Receive data set.

2. Receive data into the Receive data set via TSO RECEIVE.

3. Starts TSP BLGLOAD1.

    a. Call user exit BLGUT3WT to:

        1) Read the LOGSAVE record.

        2) Verify Receive database status.

        3) Verify LOGSAVE record status enable.

        4) Put LOGSAVE record information in the TSCA variable data area.

        5) Wait time as specified in LOGSAVE record, then return to TSP.

    b. Call user exit BLGUT3EX to:

        1) Read TSCA variable data area.

        2) If synchronization is set on:

            a) Obtain sequence number from the SDIDS.

            b) Add 1 to the sequence number.

            c) Verify that the sequence number is the same as in the Receive data set.

        3) Convert data from the Receive data set to the Information/Management database, taking filtering criteria into consideration.

        4) If synchronization is set on, store the sequence number in the SDIDS.

4. Submits next Send job BLMSASDA on the send system.

5. Sends a notification to a specified user ID.

6. The Send function is complete.

## 8.4 Using the Automatic Log Save Facility

Automatic Log Save provides us with the facility to mirror data from one Information/Management database to another. We can use this technology to assist in the following scenarios.

### 8.4.1 Automatic Backup

The Automatic Log Save facility can be used for automatic backup on the same or a different system. Then, a customer no longer has to run a full backup every night. Depending on the size of the database and the number of problems and changes happening per day, a customer might decide to run a full backup every week with Automatic Log Save to offload the SDLDS every day.

The Automatic Log Save does not require a Receive database and in the automatic backup setup the Receive database will not be defined.

This scenario depends very much on the time it will take to restore your database if something goes wrong, and whether you can afford the time in an online day.

In the the worst scenario, we are on day 7 and have to restore from the previous seven days′ backup with six days of SDLDS.

### 8.4.2 Hot Standby

The Automatic Log Save facility can also be used as a hot standby system, where in an emergency situation you could swap to your mirrored system. Using the mirrored system as your production until you have restored or fixed the original production system.

## 8.4.3 Reporting

The mirror database can be kept up to date with the production database via the Automatic Log Save utility. The mirror database can then be used for reporting, thereby taking all reporting away from your main database and improving performance for interactive users, as seen in Figure 63.

RFTs can require extensive I/O both to the SDDS and the SDIDS and to CPU resources. Moving RFTs to another mirror database can improve enqueues being experienced on the SDIDS and SDDS on the production database, as well as giving RFTs a better performance.

If CPU is a bottleneck, a mirror database on the same system can reduce overall performance. However, moving the mirror database to another system can improve the overall response times experienced.

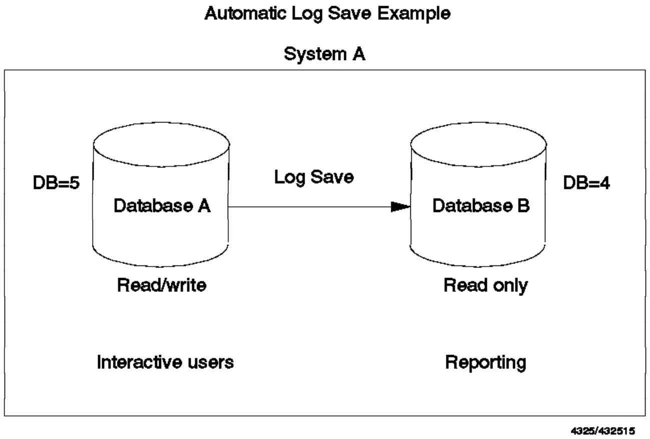


*Figure 63. Automatic Log Save Example*

## 8.4.4 Cross Pollination

In the following scenario, the customer would like to maintain two separate databases at two separate sites (site A and site B). Problems written to the database at site A must be reflected in the database image at site B, and vice versa. Images must stay in line with their corresponding originals.

The solution is Automatic Log Save and mirroring on both systems, each updating a read only database on the other system from where users could see the other system′s data. For an illustration see Figure 64 on page 80.

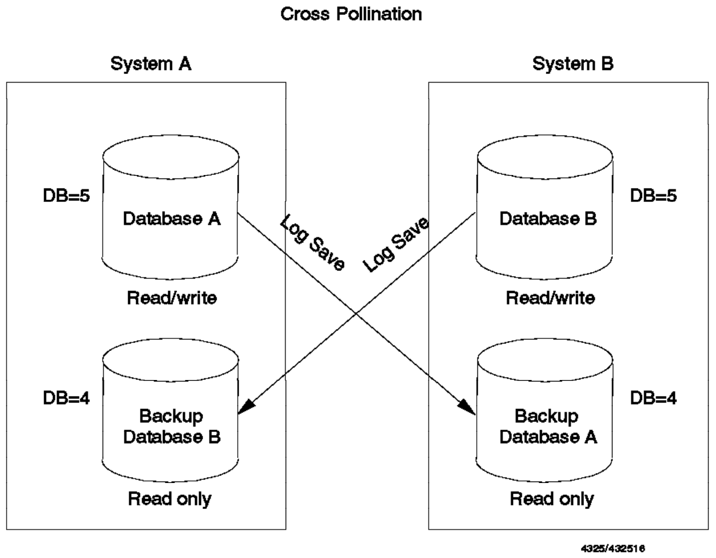


*Figure 64. Cross Pollination*

From System A you would have update access to Database A and read access to Database B and from System B you can update Database B and read access to Database A.

The session parameter reflects both the write and read database at the same time. Through TSPs and automation we could implement dual searches and even cross pollination (meaning that data from the read/only database could be used to create records in the read/write, local database, with a copy operation for example).

### 8.4.5 Mirror Frequency

The frequency you decide on for mirroring can be very important since it can have an impact on your performance. The following factors have to be taken into consideration:

- Time interval in minutes set on the LOGSAVE record.
- Performance constraints you might have in your current environment.
- How long does it take to:
  - Offload of the SDLDS.
  - Build the Receive job.
  - Route and process the Receive job.
  - Route the job to start the next Send.
- Your network configuration and the amount of traffic it carries.
- Whether you are synchronizing the Automatic Log Save Send and Receive functions.
- Can you afford an exclusive enqueue on the SDLDS for the read of 20 records? The REC=20 and the %FULL= parameters on user exit BLGUT4EX enables you to control how long the exclusive enqueue is held.

## 8.5 Planning Activities

At the start, it is a good idea to spend some time considering precisely what your objectives are, as this will influence the tasks that are involved. The following list provides some questions that will help:

1. Is the hot standby going to be running on this system, or on a remote system?

   If it is on a remote system, then such a remote system might be considered the disaster recovery database too.

2. What naming standards are going to be used for the different databases?
3. What time interval is there going to be between each iteration of Log Save?
4. How are the Log Save procedure jobs going to be administered? Are they going to be managed by the Information/Management Administrator, or is a job scheduler going to be used?
5. How will Log Save be enabled, disabled, or re-enabled?
6. Which user ID is to receive notification of the success or failure of the Log Save jobs?
7. Do you want to synchronize the Send and Receive database?
8. Will the same user ID run the Send and Receive database?
9. Do you want to filter out any records?

   Please note that if the LOGSAVE record is updated on the prime database, then this will be ignored by Log Save processing.

10. How many generations will be retained in the GDG?
11. How many records in the SDLDS should be held by the enqueue which Log Save holds?
12. How will implementing Log Save affect existing housekeeping procedures?

## 8.6 How to Implement Log Save

The checklist that follows describes the steps to implement the Automatic Log Save:

1. Set up the Send database.

   - Verify that the Send database is running with an SDLDS.
   - Issue the FREE command against the SDDS and SDIDS to prevent new records being created until the copies of these files are in place.
   - Create hot standby. SDDS and SDIDS files, either using IDCAMS, or ADRDSSU. The utility used will depend on whether the second database is local or not. If it is remote, then an ADRDSSU logical dump and restore may be the fastest method.

     If the standby system is local, then the panels and dictionary can be shared. However, we recommend that the SDDS and SDIDS for the Hot Standby be on separate DASD.

   - Issue the REALLOC command against the SDDS and SDIDS.
   - Create the LOGSAVE record for the Send database.

- Create a GDG for the Send data set.
- Prepare the Send JCL, BLMSASD,BLMSASDA and BLMSASDE, as described in 8.6.2, “How to Tailor the JCL.”
- Update TSP BLGDUMP1 if you wish to change the default value of RECS=20, which affects how many records are processed each time an enqueue is obtained on the SDLDS.

2. Set up the Receive database (hot standby database).
   - Define a session parameter for the Receive database with no SDLDS.
   - Prepare the Receive JCL, BLMSARV and BLMSAL1 as described in 8.6.2, “How to Tailor the JCL.” The JCL BLMSASDA needs to be available at the Receive end too, so if this is a different MVS, the JCL will need to be copied over.
   - Create the LOGSAVE record for the Receive database.
3. Submit BLMSASD on the MVS/ESA system where the Send database resides.

## 8.6.1 Description of Log Save JCL

There are a total of five JCL members involved, as follows:

**BLMSASD** This is the Send job used to initiate the Automatic Log Save Facility, and runs on the MVS image where your Send database resides.

**BLMSARV** This is the first part of the JCL for the receive.

**BLMSAL1** This is the second part of the JCL for the receive.

**BLMSASDA** This is similar to BLMSASD, but is submitted by BLMSARV after a successful receive.

**BLMSASDE** If there has been a successful send, but the receive has failed, then this job needs to be submitted manually, on the send MVS image, to initiate a receive in those circumstances.

## 8.6.2 How to Tailor the JCL

Copy each of the JCL members described below from SBLMSAMP into your JCL library, make the changes described but don′t submit any jobs at this time.

## 8.6.3 Preparing BLMSASD

Invoke ISPF edit to modify your copy of this job, and make the following changes:

1. Type `PROFILE`, and verify that your member has *PACK OFF*. If it hasn′t, change it because *PACK ON* will cause problems with the automatic submission of the receive.
2. Change the job card to your local standards.
3. Change the `/*ROUTE PRINT` to the node and user ID that you wish to receive the job output, or delete the card completely if you wish to have the output kept locally.
4. In step OFFLD, change the session parameter suffix in your SYSIN to match that of the sending database, then change the data set names for the following DDnames to those for your installation:
   - STEPLIB - Include the ISPF load library, if it is not in the MVS linklist.
   - ISPPROF

- ISPPLIB - Include the library that contains panel BLGISPFD, or the job will fail.
- ISPTLIB
- ISPMLIB
- ISPSLIB
- ISPLLIB
- BLGTRACE
- SYSPROC
- SYSTSPRT

5. Change the *sample message*, and the *node.userid* to receive the messages, in the following steps:

   - GOODOFF
   - FAILOFF
   - GOODSUB

6. Change the JCL data set name to that of your tailored JCL in the following steps:

   - COPY1
   - COPY2

7. In step SAVESND:

   - Change *userid.dsname1* on the SYSUT1 and SYSUT2 DDnames to the one you have selected in the LOGSAVE record.
   - change *userid.gdgname1* to the one you have defined earlier.

8. In step XMITSND, change *userid.dsname1* to the data set name in the LOGSAVE record.

9. In step SUBSND, change the data set name of your tailored JCL library.

### 8.6.4 Preparing BLMSASDA

Invoke ISPF edit to modify your copy of this job, then make the same changes as in 8.6.3, "Preparing BLMSASD" on page 82. In addition, you need to make a further change to the DELETE step. This step needs to have the *userid.dsname1* changed to the data set name you have specified for in LOGSAVE record for the sending database.

### 8.6.5 Preparing BLMSASDE

Invoke ISPF edit to modify your copy of this job, and make the changes as described 8.6.3, "Preparing BLMSASD" on page 82.

### 8.6.6 Preparing BLMSARV

Invoke ISPF edit to modify your copy of this job, and make the following changes:

1. Type PROFILE, and verify that your member has *PACK OFF*. If it hasn't, change it because *PACK ON* will cause problems with the automatic submission of the receive.

2. The DELETE step needs to have the *userid.dsname1* changed to the data set name you have specified in your LOGSAVE record on the Receive database.

### 8.6.7 Preparing BLMSAL1

Invoke ISPF edit to modify your copy of this job, and make the following changes:

1. Type profile, and verify that your member has *PACK OFF.* If it hasn′t, then change it because *PACK ON* will cause problems with the automatic submission of the receive.
2. The RCVSND step needs to have the *userid.dsname1* changed to the data set name you have specified in your LOGSAVE record on the Receive database. This should match the name in the DELETE step of BLMSARV.
3. In the JOIN step, review the space and unit parameters, and modify them if your installation requires it.
4. In step ONLOAD, make the ISPF changes described in 8.6.3, "Preparing BLMSASD" on page 82.
5. Change the *sample message* and *node.userid* to receive the messages, in the following steps:
   - GOODON
   - FAILON
   - GOODSUB
6. In step SUBSND, change the JCL data set name to the JCL library on your the MVS image containing the Receive database.

## 8.7 Hot Standby Using NetView FTP

The NetView File Transfer Program (FTP) V2 MVS can be used with the Automated Log Save facility of Information/Management Version 6 Release 1 to provide an alternative hot standby option for the customer on his problem, change and configuration systems.

This support is provided by the following five jobs, which will be discussed in detail later in this section.

The two main jobs are:

- SEND - a job invoking Information/Management in batch to run a TSP to offload the SDLDS and, if successful, builds and submits a batch job to run on the receive system to load the created Send data set into the receive database.

  This replaces job BLMSASD and BLMSASDA. For more information on BLMSASD and BLMSASDA please see the *Information/Management Program Administration Guide and Reference* Guide, SC34-4341.

  SEND replaces the TRANSMIT steps with one FTP step. Otherwise, all other steps are the same.

- RECEIVE - a job performing an Information/Management Automatic Log Save receive.

  This replaces BLMSARV and BLMSAL1. For more information on BLMSARV and BLMSAL1 please see the *Information/Management Program Administration Guide and Reference* Guide, SC34-4341.

The three smaller jobs are:

- GOODSUB

This job takes notification step in BLMSARV and makes it a job.

- FAILSUB

  This job is the same as GOODSUB, but the job fails and abends.

- RESEND

  This job replaces BLMSASDE. Again some transmit steps are removed, and a FTP step is added.

### 8.7.1 Logic of the JCL

There are basically two jobs, a Send job and a Receive job. In addition, there is a piece of JCL which is for a RESEND job, which is used if a previously created Send data set was not received due to some failure.

### 8.7.2 SEND

An example of the SEND, GOODSUB and FAILSUB jobs can be seen in Figure 65 on page 86 to Figure 67 on page 89.

1. DELETE: The previous offload get deleted.
2. OFFLOAD: Information/Management is invoked in batch to run a TSP(BLGDUMP1) to OFFLOAD the SDLDS.
3. Condition code checking:
   - GOODSUB: A successful confirmation will be sent to a user ID.
   - FAILSUB: An error message will be sent to a user ID. It could be that an error occurred or the LOGSAVE record status is set to DISABLED. Please refer to the job output for more information.
4. SAVESND: Make a backup copy of the Send data set to a GDG. The GDG can be defined using sample JCL BLGDG1.
5. FTPDCB: Submits a batch job to run on the Receive system to load the created Send data set into the Receive data set.
6. Condition code checking:
   - GOODSUB: A successful confirmation will be sent to a user ID.
   - FAILSUB: An error message will be sent to a user ID.

Of the steps above, only the FTPDCB step is new. The logical difference from Automatic Log Save is the OFFLOAD step, because it presumes a data set was left on the system.

```
//BLMSEND  JOB (nnnnnn,nnnn,nnn,,,N),’userid’,
//         MSGLEVEL=(1,1),MSGCLASS=T,
//         CLASS=A,NOTIFY=userid,USER=userid
//* DEFINE ROUTING INFORMATION.
//*-------------------------------------------------------------*
//*                                                             *
//*           LICENSED MATERIALS - PROPERTY OF IBM              *
//*                                                             *
//*           5695-171 (C) COPYRIGHT IBM CORP. 1994.            *
//*                                                             *
//*           SEE COPYRIGHT INSTRUCTIONS                        *
//*                                                             *
//*-------------------------------------------------------------*
//* THIS JOB INVOKES INFO/MANAGEMENT IN BATCH TO RUN A TSP TO   */
//* OFFLOAD THE SDLDS AND IF SUCCESSFUL BUILDS AND SUBMITS A    */
//* BATCH JOB TO RUN ON THE RECEIVE SYSTEM TO LOAD THE CREATED  */
//* SEND DATA SET INTO THE RECEIVE DATABASE.  DO NOT USE THIS   */
//* JCL IF YOU ARE PROCESSING A PREVIOUSLY CREATED SEND DATA    */
//* SET THAT WAS NOT RECEIVED DUE TO SOME FAILURE.  IN THAT     */
//* CASE, YOU SHOULD USE JOB BLMSASDE WHICH PROCESSES THE       */
//* EXISTING SEND DATA SET AND THEN STARTS NORMAL OFFLOAD/LOAD  */
//* PROCESSING.                                                 */
//*                                                             */
//* THIS JCL IS SUBMITTED MANUALLY ON THE SYSTEM WHERE THE      */
//* AUTOMATIC LOG SAVE SEND DATABASE RESIDES.                   */
//*                                                             */
//* THE OFFLOADED SDLDS IS STORED IN THE ”SEND” DATASET.        */
//* TSO TRANSMIT IS USED TO CONVERT THE DATA INTO FB=80 FORMAT  */
//* AND THEN EACH RECORD IS SPLIT INTO TWO 80-BYTE RECORDS      */
//* WITH A BLANK AS THE FIRST CHARACTER.  THIS DATA IS          */
//* INCLUDED IN-STREAM IN THE BUILT RECEIVE JCL.  THE           */
//* RE-FORMATTING IS REQUIRED SO THAT THE IN-STREAM DATA WILL   */
//* NOT BE INADVERTENTLY DELIMITED BY A VALUE IN THE DATA.      */
//*                                                             */
//* IF A TRANSMIT ERROR OCCURS, ASSUME THAT IT IS BECAUSE THE   */
//* SEND DATA SET IS EMPTY AND SUBMIT A JOB TO PERFORM THE      */
//* NEXT SEND (SUBSND STEP).                                    */
//*                                                             */
//* THE RECEIVE JCL FORMATS THE TWO 80-BYTE RECORDS BACK INTO   */
//* A SINGLE 80-BYTE RECORD AND USES TSO RECEIVE TO PLACE       */
//* THE IN-STREAM DATA INTO THE RECEIVE DATA SET AND THEN       */
//* PERFORMS THE ONLOAD OF THE SDLDS.                           */
//*-------------------------------------------------------------*/
//*    UPDATE THE FOLLOWING WITH UPPERCASE TEXT:                */
//*                                                             */
//*  STEP               INSTRUCTIONS                            */
//* ============================================================*/
//*  ------      - CHANGE JOBCARD TO YOUR LOCAL STANDARDS.      */
//*  ------      - CHANGE THE node.userid ON THE /*ROUTE PRINT  */
//*                CARD TO THE NODE AND ID THAT WILL RECEIVE    */
//*                THE JOB OUTPUT.                              */
//*  OFFLD       - CHANGE THE DATA SET NAMES FOR DD NAMES       */
//*                STEPLIB, ISPPROF, ISPPLIB, ISPTLIB, ISPMLIB, */
//*                ISPSLIB, BLGTRACE, ISPLLIB, SYSPROC, AND     */
//*                SYSTSPRT TO CORRESPOND TO THE DATA SET NAMES */
//*                AT YOUR INSTALLATION.                        */
//*               *DD NAMES ISPXXXX ARE THE LIBRARIES YOU USE   */
//*                TO ACCESS ISPF.                              */
//*               *DD NAME BLGTRACE CAN BE USED TO TRACE THE    */
//*                FLOW OF THE TSP WHICH PERFORMS THE OFFLOAD   */
//*                OF THE SDLDS FOR INFORMATIONAL OR DEBUG      */
//*                PURPOSES.                                    */
```

*Figure 65 (Part 1 of 3). SEND Job Example*

```
//*              - CHANGE THE VALUE FOR SESS(xx) TO THE SESSION */
//*                MEMBER FOR YOUR SEND DATABASE.               */
//*  GOODOFF     - CHANGE sample message TO YOUR MESSAGE FOR    */
//*                A SUCCESSFUL EXECUTION OF THE SDLDS          */
//*                OFFLOAD (CONDITION CODE == 0000).            */
//*              - CHANGE node.userid TO THE DESTINATION FOR    */
//*                THIS MESSAGE.                                */
//*  FAILOFF     - CHANGE sample message TO YOUR MESSAGE FOR    */
//*                A RETURN CODE OF NOT 0 FROM THE OFFLD        */
//*                STEP.                                        */
//*              - CHANGE node.userid TO THE DESTINATION FOR    */
//*                THIS MESSAGE.                                */
//*  SAVESND     - CHANGE userid.dsname1 TO THE NAME OF THE SEND*/
//*                DATASET IN THE LOGSAVE RECORD.               */
//*              - CHANGE userid.gdgname1 TO THE NAME OF THE GDG*/
//*                FOR THE SEND DATA SET.                       */
//*              - CHANGE THE DCB ATTRIBUTES FOR THE GDG        */
//*                GENERATION TO MATCH THOSE USED FOR THE       */
//*                SEND DATA SET.                               */
//*              - CHANGE userid.dsname1 IN THE DCB PARAMETER   */
//*                TO MATCH DATASET ON SYSUT1                   */
//*  GOODSUB     - CHANGE sample message TO YOUR MESSAGE FOR    */
//*                A SUCCESSFUL SUBMISSION OF THE RECEIVE       */
//*                JCL.                                         */
//*              - CHANGE node.userid TO THE DESTINATION FOR    */
//*                THIS MESSAGE.                                */
//*  FTPDCB      - CHANGE userid TO THE USERID's TO BE USED     */
//*-------------------------------------------------------------*/
//DELETE   EXEC PGM=IEFBR14
//*
//DD1      DD DSN=userid.OFFLOAD,DISP=(MOD,DELETE),
//         UNIT=SYSDA,SPACE=(TRK,(1,1))
//**************************************************************/
//* START INFO/MANAGEMENT IN BATCH AND RUN TSP TO PERFORM THE   */
//* OFFLOAD OF THE SDLDS.                                       */
//**************************************************************/
//OFFLD   EXEC  PGM=IKJEFT01,DYNAMNBR=25,REGION=4096K
//STEPLIB DD DSN=userid.TEST.LOAD,DISP=SHR
//        DD DSN=userid.APF.LOAD,DISP=SHR
//        DD DSN=INFOPROJ.V5BASE.LOAD,DISP=SHR
//ISPPROF DD DSN=userid.BATCH.PROFILE,DISP=SHR
//ISPPLIB DD DSN=SYS1.ISR.ISRPENU,DISP=SHR
//        DD DSN=SYS1.ISP.ISPPENU,DISP=SHR
//*       DD DSN=IP01.ISPPLIB,DISP=SHR
//        DD DSN=INFOPROJ.MENU.PANELS,DISP=SHR
//*       DD DSN=LCG.JOY6107.ABLMSAMP,DISP=SHR
//ISPSLIB DD DSN=SYS1.ISR.ISRSENU,DISP=SHR
//        DD DSN=SYS1.ISP.ISPSLIB,DISP=SHR
//ISPLOG  DD SYSOUT=O,
//           DCB=(RECFM=VA,LRECL=125,BLKSIZE=129)
//SYSPROC DD DSN=userid.CLIST,DISP=SHR
//        DD DSN=SYS1.ISR.ISRCLIB,DISP=SHR
//ISPTLIB DD DSN=SYS1.ISR.ISRTLIB,DISP=SHR
//        DD DSN=SYS1.ISP.ISPTENU,DISP=SHR
//ISPMLIB DD DSN=SYS1.ISR.ISRMENU,DISP=SHR
//        DD DSN=SYS1.ISP.ISPMENU,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//BLGTRACE DD SYSOUT=*
//SYSTSIN  DD   *
  ISPSTART PGM(BLGINIT) PARM(SESS(SE) TSP(BLGDUMP1))
//**************************************************************/
```

*Figure 65 (Part 2 of 3). SEND Job Example*

```
//*   TRANSMIT A SUCCESSFUL MESSAGE TO A userid                 */
//**************************************************************/
//GOODOFF  EXEC  PGM=IKJEFT01,COND=(0,NE,OFFLD)
//SYSTSPRT DD    SYSOUT=*
//*
//GOODOFF  DD    *
This is a sample message to indicate that the batch invocation to
run TSP BLGDUMP1 completed with a code of 0.
//SYSTSIN  DD    *
  TRANSMIT CARVM3.userid DDNAME(GOODOFF) NOLOG NONOTIFY
//**************************************************************/
//*   TRANSMIT A FAILURE MESSAGE TO A userid                    */
//**************************************************************/
//FAILOFF  EXEC  PGM=IKJEFT01,COND=((0,EQ,OFFLD),EVEN)
//SYSTSPRT DD    SYSOUT=*
//*
//FAILOFF  DD    *
This is a sample message to indicate that the batch invocation to
run TSP BLGDUMP1 completed with a code of greater than 0 which means
an error occurred or the LOGSAVE record status is set to DISABLED.
It should refer the user to the job output for more information.
//SYSTSIN  DD    *
  TRANSMIT CARVM3.userid DDNAME(FAILOFF) NOLOG NONOTIFY
//**************************************************************/
//* MAKE A BACKUP COPY OF THE SEND DATA SET TO A GDG.           */
//* THE GDG CAN BE DEFINED USING SAMPLE JCL BLGDG1.             */
//**************************************************************/
//SAVESND  EXEC PGM=IEBGENER,COND=(0,NE,OFFLD)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
//SYSUT1   DD DSN=userid.dsname1,DISP=(OLD,KEEP)
//*
//SYSUT2   DD DSN=userid.gdgname1(+1),DISP=(NEW,CATLG,DELETE),
//         UNIT=SYSDA,SPACE=(TRK,(150,50),RLSE),
//         DCB=(userid.dsname1,RECFM=VB,BLKSIZE=8206,LRECL=8202)
/*
//*****************************************************************
//FTPDCB   EXEC PGM=DVGIFBI,REGION=1M
//DVGLOG   DD   SYSOUT=*
//SYSIN    DD   *
FUNCTION=ADD
XMODE=TO
WAIT=YES
REQUEUE=YES
REMNODE=SYSTEM
SNOTIFY=userid
SPTJOBLB=′userid.JCL′
SPTJOBOK=RECEIVE
SPTJOBOK=GOODSUB
SPTJOBNO=FAILSUB
CLASS=0
COMPMODE=ADAPT
PRTY=0
SSRVMODE=CONT
RSRVMODE=CONT
SFILEID=′userid.OFFLOAD′
RFILEID=′userid.ONLOAD′
RTDISP=CAT
RSECURP=(userid,PASSWRD)
SSECURP=(userid,PASSWRD)
/*
```

*Figure 65 (Part 3 of 3). SEND Job Example*

```
//BLMSAMSG JOB CLASS=A,MSGCLASS=R,MSGLEVEL=(1,1),NOTIFY=userid
//*  GOODSUB     - CHANGE sample message TO YOUR MESSAGE FOR    */
//*                A SUCCESSFUL SUBMISSION OF THE RECEIVE       */
//*                JCL.                                         */
//*              - CHANGE node.userid TO THE DESTINATION FOR    */
//*                THIS MESSAGE.                                */
//GOODSUB  EXEC  PGM=IKJEFT01,
//         COND=((0,NE,OFFLD),(0,NE,SAVESND))
//SYSTSPRT DD    SYSOUT=*
//*
//GOODSUB  DD    *
This is a sample message to indicate that the job to perform the
Automatic Log Save Receive was submitted successfully.
//SYSTSIN  DD    *
  TRANSMIT node.userid DDNAME(GOODSUB) NOLOG NONOTIFY
```

*Figure 66. GOODSUB Job Example*

```
//BLMSAMSG JOB CLASS=A,MSGCLASS=R,MSGLEVEL=(1,1),NOTIFY=userid
//***********************************************************/
//*   TRANSMIT A FAILURE MESSAGE TO A USERID                */
//*  FAILSUB     - CHANGE sample message TO YOUR MESSAGE FOR   */
//*                AN UNSUCCESSFUL SUBMISSION OF THE RECEIVE   */
//*                JCL.                                        */
//*              - CHANGE node.userid TO THE DESTINATION FOR   */
//*                THIS MESSAGE.                               */
//***********************************************************/
//FAILTRNS EXEC  PGM=IKJEFT01,
//    COND=((0,EQ,ONLOAD),EVEN)
//SYSTSPRT DD    SYSOUT=*
//*
//FAILTRNS DD    *
This is a sample message to indicate that the NFT invocation to
copy the Send data set to the Receive system failed.
//*
//SYSTSIN  DD    *
  TRANSMIT system.userid DDNAME(FAILTRNS) NOLOG NONOTIFY
//*END OF JOB
```

*Figure 67. FAILSUB Job Example*

### 8.7.3 RECEIVE

An example of the Receive job can be seen in Figure 68 on page 90.

1. ONLOAD: Loads the offload data set in the system.
2. Condition code checking:
   - GOODON: A successful confirmation will be sent to a user ID.
   - FAILON: An error message will be sent to a user ID.
3. DELETE: The previous onload gets deleted.
4. SUBSND: Submits a batch job to the internal reader. The job will kick off the next send, which is routed to the sending system.
5. Condition code checking:
   - GOODSUB: A successful confirmation will be sent to a user ID.

- Initiate the SEND job again.

```
//BLMSARV  JOB (nnnnnn,nnnn,nnn,,,N),'userid',
//         MSGLEVEL=(1,1),MSGCLASS=T,
//         CLASS=A,NOTIFY=userid,USER=userid
//*            USER=userid,PASSWORD=PASSWRD
//*
//* DEFINE ROUTING INFORMATION.
//*ROUTE XEQ CARMVS1
//*
//*-----------------------------------------------------------*
//*                                                           *
//*           LICENSED MATERIALS - PROPERTY OF IBM            *
//*                                                           *
//*           5695-171 (C) COPYRIGHT IBM CORP. 1994.          *
//*                                                           *
//*           SEE COPYRIGHT INSTRUCTIONS                      *
//*                                                           *
//*-----------------------------------------------------------*
//* THIS JOB PERFORMS AN INFO/MANAGEMENT AUTOMATIC LOG SAVE     */
//* RECEIVE.                                                    */
//*                                                             */
//* IT REFORMATS THE IN-STREAM SEND DATA SET DATA INTO          */
//* TSO TRANSMIT FORMAT, RECEIVES THE DATA INTO THE RECEIVE     */
//* DATA SET (USING TSO RECEIVE), AND STARTS INFO/MANAGEMENT   */
//* IN BATCH TO PERFORM THE DATABASE ONLOAD.  IF THE ONLOAD     */
//* IS SUCCESSFUL, IT SUBMITS A BATCH JOB THAT IS ROUTED TO THE */
//* SEND SYSTEM TO PERFORM THE NEXT SEND OPERATION.             */
//*-----------------------------------------------------------*/
//*    UPDATE THE FOLLOWING WITH UPPERCASE TEXT:                */
//*                                                             */
//*  STEP               INSTRUCTIONS                            */
//* ==========================================================*/
//*  ------      - CHANGE JOBCARD TO YOUR LOCAL STANDARDS.      */
//*                THIS JOB WILL BE ROUTED TO THE SYSTEM WHERE  */
//*                THE INFO/MANAGEMENT RECEIVE DATABASE         */
//*                RESIDES - YOU MAY NEED USER AND PASSWORD     */
//*  ------      - CHANGE node on /*ROUTE XEQ CARD TO THE NODE  */
//*                WHERE THE JOB IS TO RUN.                     */
//*  ------      - CHANGE THE node.userid ON THE /*ROUTE PRINT  */
//*                CARD TO THE NODE AND ID THAT WILL RECEIVE    */
//*                THE JOB OUTPUT.                              */
//*  DELETE      - CHANGE userid.dsname2 TO THE NAME OF THE     */
//*                RECEIVE DATA SET THAT IS SPECIFIED IN THE    */
//*                LOGSAVE RECORD.                              */
//*                THIS NAME MUST MATCH THE DATA SET NAME       */
//*                SPECIFIED IN THE RCVSND STEP IN THE BLMSAL1  */
//*                JCL.                                         */
//*-----------------------------------------------------------*/
//************************************************************/
//* DELETE THE DATA SET THAT CONTAINS THE RECEIVE DATA SET      */
//* FROM THE PREVIOUS INVOCATION OF RECEIVE.                    */
//************************************************************/
//ONLOAD   EXEC  PGM=IKJEFT01,DYNAMNBR=25,REGION=4096K
//STEPLIB DD DSN=userid.TEST.LOAD,DISP=SHR
//        DD DSN=userid.APF.LOAD,DISP=SHR
//        DD DSN=INFOPROJ.V5BASE.LOAD,DISP=SHR
//ISPPROF DD DSN=userid.BATCH.PROFILE,DISP=SHR
```

*Figure 68 (Part 1 of 3). RECEIVE Job Example*

```
//ISPPLIB DD DSN=SYS1.ISR.ISRPENU,DISP=SHR
//        DD DSN=SYS1.ISP.ISPPENU,DISP=SHR
//*       DD DSN=IPO1.ISPPLIB,DISP=SHR
//        DD DSN=INFOPROJ.MENU.PANELS,DISP=SHR
//*       DD DSN=LCG.JOY6107.ABLMSAMP,DISP=SHR
//ISPSLIB DD DSN=SYS1.ISR.ISRSENU,DISP=SHR
//        DD DSN=SYS1.ISP.ISPSLIB,DISP=SHR
//ISPLOG  DD SYSOUT=O,
//           DCB=(RECFM=VA,LRECL=125,BLKSIZE=129)
//SYSPROC DD DSN=userid.CLIST,DISP=SHR
//        DD DSN=SYS1.ISR.ISRCLIB,DISP=SHR
//ISPTLIB DD DSN=SYS1.ISR.ISRTLIB,DISP=SHR
//        DD DSN=SYS1.ISP.ISPTENU,DISP=SHR
//ISPMLIB DD DSN=SYS1.ISR.ISRMENU,DISP=SHR
//        DD DSN=SYS1.ISP.ISPMENU,DISP=SHR
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//BLGTRACE DD SYSOUT=*
//SYSTSIN  DD   *
  ISPSTART PGM(BLGINIT) PARM(SESS(RV) TSP(BLGLOAD1))
//**************************************************************/
//*   TRANSMIT A SUCCESSFUL MESSAGE TO A USERID                */
//**************************************************************/
//GOODON   EXEC  PGM=IKJEFT01,
//         COND=(0,NE,ONLOAD)
//SYSTSPRT DD    SYSOUT=*
//*
//GOODON   DD    *
This is a sample message to indicate that the batch invocation to
run TSP BLGLOAD1 completed with a code of 0.
//SYSTSIN  DD    *
  TRANSMIT CARVM3.userid DDNAME(GOODON) NOLOG NONOTIFY
//**************************************************************/
//*   TRANSMIT A FAILURE MESSAGE TO A USERID                   */
//**************************************************************/
//FAILON   EXEC  PGM=IKJEFT01,
//    COND=((0,EQ,ONLOAD),EVEN)
//SYSTSPRT DD    SYSOUT=*
//*
//FAILON   DD    *
This is a sample message to indicate that the batch invocation to
run TSP BLGLOAD1 completed with a code of greater than 0 which means
an error occurred or was not attempted due to earlier errors.
It should direct the user to check the job output for more
information.
//*
//SYSTSIN  DD    *
  TRANSMIT CARVM3.userid DDNAME(FAILON) NOLOG NONOTIFY
//*
//DELETE   EXEC PGM=IEFBR14,
//    COND=(0,NE,ONLOAD)
//*
//DD1      DD DSN=userid.ONLOAD,DISP=(MOD,DELETE),
//         UNIT=SYSDA,SPACE=(TRK,(1,1))
```

*Figure 68 (Part 2 of 3). RECEIVE Job Example*

```
//**************************************************************/
//* SUBMIT A JOB TO THE INTERNAL READER.  THE JOB WILL KICK    */
//* OFF THE NEXT SEND (IT IS ROUTED TO THE SENDING SYSTEM).    */
//**************************************************************/
//SUBSND   EXEC PGM=IEBEDIT,
//    COND=(0,NE,ONLOAD)
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DDNAME=IEFRDER
//SYSUT2   DD SYSOUT=(A,INTRDR),DCB=BLKSIZE=80
//SYSIN    DD *
     EDIT NOPRINT
//IEFRDER  DD DSN=userid.NVFTP.JCL(SEND),DISP=SHR
//**************************************************************/
//* IF THE ONLOAD WAS SUCCESSFUL AND THE NEXT SEND WAS         */
//* SUBMITTED SUCCESSFULLY, SEND A SUCCESS MESSAGE TO A USER.  */
//**************************************************************/
//GOODSUB  EXEC  PGM=IKJEFT01,
//         COND=((0,NE,ONLOAD),
//         (0,NE,SUBSND))
//SYSTSPRT DD    SYSOUT=*
//*
//GOODSUB  DD    *
This is a sample message to indicate that the job to perform the
next Automatic Log Save Send was submitted successfully.
//SYSTSIN  DD    *
  TRANSMIT node.userid DDNAME(GOODSUB) NOLOG NONOTIFY
//*
//*END OF JOB
```

*Figure 68 (Part 3 of 3). RECEIVE Job Example*

## 8.7.4 RESEND

If the Send data set failed we recommend that you use the RESEND to send the data set to the Receive system.

```
//RESEND   JOB (nnnnnn,nnn,nnn,,,N),′userid′,
//         MSGLEVEL=(1,1),MSGCLASS=T,
//         CLASS=A,NOTIFY=userid,USER=userid
//* DEFINE ROUTING INFORMATION.
//*-------------------------------------------------------------*
//*                                                             *
//*           LICENSED MATERIALS - PROPERTY OF IBM              *
//*                                                             *
//*           5695-171 (C) COPYRIGHT IBM CORP. 1994.            *
//*                                                             *
//*           SEE COPYRIGHT INSTRUCTIONS                        *
//*                                                             *
//*-------------------------------------------------------------*
//* THIS JOB TAKES AN SDLDS OFFLOADED DATASET AND SUBMITS A     */
//* BATCH JOB TO RUN ON THE RECEIVE SYSTEM TO LOAD THE CREATED  */
//* SEND DATA SET INTO THE RECEIVE DATABASE.  USE THIS         */
//* JCL IF YOU ARE PROCESSING A PREVIOUSLY CREATED SEND DATA    */
//* SET THAT WAS NOT RECEIVED DUE TO SOME FAILURE.              */
//* IT THEN STARTS NORMAL OFFLOAD/LOAD  PROCESSING              */
//*                                                             */
//* THIS JCL IS SUBMITTED MANUALLY ON THE SYSTEM WHERE THE      */
//* AUTOMATIC LOG SAVE SEND DATABASE RESIDES.                   */
//*                                                             */
//* THE OFFLOADED SDLDS IS STORED IN THE ″SEND″ DATASET.        */
//* NetView FTP IS USED TO TRANSMIT THE DATA.                   */
//*                                                             */
//* IF A TRANSMIT ERROR OCCURS, SEE THE NVFTP CONSOLE AND       */
//* NVFTP MESSAGES GUIDE FOR ERROR PROCESSING PROCEDURES.       */
//*                                                             */
//* THE RECEIVE JCL PERFORMS THE ONLOAD OF THE SDLDS.           */
//* IF A RECEIVE ERROR OCCURS, SEE THE INFO/MANAGEMENT          */
//* DOCUMENTATION FOR ERROR PROCESSING PROCEDURES.              */
//*-------------------------------------------------------------*/
//*    UPDATE THE FOLLOWING WITH UPPERCASE TEXT:                */
//*                                                             */
//*  STEP               INSTRUCTIONS                            */
//* ============================================================*/
//*  ------      - CHANGE JOBCARD TO YOUR LOCAL STANDARDS.      */
//*  ------      - CHANGE THE node.userid ON THE /*ROUTE PRINT  */
//*                CARD TO THE NODE AND ID THAT WILL RECEIVE    */
//*                THE JOB OUTPUT.                              */
//*  GOODOFF     - CHANGE sample message TO YOUR MESSAGE FOR    */
//*                A SUCCESSFUL EXECUTION OF THE SDLDS          */
//*                OFFLOAD (CONDITION CODE == 0000).            */
//*              - CHANGE node.userid TO THE DESTINATION FOR    */
//*                THIS MESSAGE.                                */
//*  FAILOFF     - CHANGE sample message TO YOUR MESSAGE FOR    */
//*                A RETURN CODE OF NOT 0 FROM THE OFFLD        */
//*                STEP.                                        */
//*              - CHANGE node.userid TO THE DESTINATION FOR    */
//*                THIS MESSAGE.                                */
//*  SAVESND     - CHANGE userid.dsname1 TO THE NAME OF THE SEND*/
//*                DATASET IN THE LOGSAVE RECORD.               */
//*              - CHANGE userid.gdgname1 TO THE NAME OF THE GDG*/
//*                FOR THE SEND DATA SET.                       */
//*              - CHANGE THE DCB ATTRIBUTES FOR THE GDG        */
//*                GENERATION TO MATCH THOSE USED FOR THE       */
//*                SEND DATA SET.                               */
//*              - CHANGE userid.dsname1 IN THE DCB PARAMETER   */
//*                TO MATCH DATASET ON SYSUT1                   */
```

*Figure 69 (Part 1 of 3). RESEND Job Example*

```
//*  FTPDCB      - CHANGE node.userid TO THE DESTINATION FOR    */
//*                THIS DATASET.                                */
//*-------------------------------------------------------------*/
//DELETE   EXEC PGM=IEFBR14
//*
//DD1      DD DSN=userid.OFFLOAD,DISP=(MOD,DELETE),
//         UNIT=SYSDA,SPACE=(TRK,(1,1))
//***************************************************************/
//*   TRANSMIT A SUCCESSFUL MESSAGE TO A USERID                 */
//***************************************************************/
//GOODOFF  EXEC  PGM=IKJEFT01,COND=(0,NE,OFFLD)
//SYSTSPRT DD    SYSOUT=*
//*
//GOODOFF  DD    *
This is a sample message to indicate that the batch invocation to
run TSP BLGDUMP1 completed with a code of 0.
//SYSTSIN  DD    *
  TRANSMIT system.userid DDNAME(GOODOFF) NOLOG NONOTIFY
//***************************************************************/
//*   TRANSMIT A FAILURE MESSAGE TO A USERID                    */
//***************************************************************/
//FAILOFF  EXEC  PGM=IKJEFT01,COND=((0,EQ,OFFLD),EVEN)
//SYSTSPRT DD    SYSOUT=*
//*
//FAILOFF  DD    *
This is a sample message to indicate that the batch invocation to
run TSP BLGDUMP1 completed with a code of greater than 0 which means
an error occurred or the LOGSAVE record status is set to DISABLED.
It should refer the user to the job output for more information.
//SYSTSIN  DD    *
  TRANSMIT system.userid DDNAME(FAILOFF) NOLOG NONOTIFY
//***************************************************************/
//* MAKE A BACKUP COPY OF THE SEND DATA SET TO A GDG.           */
//* THE GDG CAN BE DEFINED USING SAMPLE JCL BLGDG1.             */
//***************************************************************/
//SAVESND  EXEC PGM=IEBGENER,COND=(0,NE,OFFLD)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
//*
//SYSUT1   DD DSN=userid.dsname1,DISP=(OLD,KEEP)
//*
//SYSUT2   DD DSN=userid.gdgname1(+1),DISP=(NEW,CATLG,DELETE),
//         UNIT=SYSDA,SPACE=(TRK,(150,50),RLSE),
//         DCB=(userid.dsname1,RECFM=VB,BLKSIZE=8206,LRECL=8202)
/*
//*******************************************************************
//FTPDCB   EXEC PGM=DVGIFBI,REGION=1M
//DVGLOG   DD   SYSOUT=*
//SYSIN    DD   *
FUNCTION=ADD
XMODE=TO
WAIT=YES
REQUEUE=YES
REMNODE=system
SNOTIFY=userid
SPTJOBLB='userid.NVFTP.JCL'
SPTJOBOK=RECEIVE
SPTJOBOK=GOODSUB
SPTJOBNO=FAILSUB
CLASS=0
COMPMODE=ADAPT
PRTY=0
```

*Figure 69 (Part 2 of 3). RESEND Job Example*

```
SSRVMODE=CONT
RSRVMODE=CONT
SFILEID=′userid.OFFLOAD′
RFILEID=′userid.ONLOAD′
RTDISP=CAT
RSECURP=(userid,ALL4YOU)
SSECURP=(userid,ALL4YOU)
/*
//*END OF JOB
```

*Figure 69 (Part 3 of 3). RESEND Job Example*

Of the steps above, only the FTPDCB step is new. The logical difference from Automatic Log Save is the OFFLOAD step, because it presumes a data set was left on the system.

### 8.7.5 NVFTP Processing

The new step that does the NVFTP processing is called FTPDCB, and is in the SEND and RESEND jobs. See Chapter 6 in the *NetView FTP V2 MVS User′s Guide*, for an example of a post-transfer job. The Send and Receive jobs are based on this post-transfer job.

### 8.7.6 New Setup Steps for NVFTP JCL

Put the new JCL where its INFO counterpart would have gone. See *Information/Management Program Administration Guide and Reference*, SC34-4341 for detail.

### 8.7.7 Problem Determination

For any NetView FTP problem please refer to the *NetView File Transfer Messages and Codes Manual*, SH12-5483, and for any Automatic Log Save Problem, please refer to the *Information/Management Program Administration Guide and Reference*, SC34-4341.

## 8.8 Implementing DB2 Extract

This section describes how to implement the DB2 Extract Facility. It should be noted at the outset that there is a significant difference between how this utility and Automatic Log Save should be used. The Automatic Log Save can be used to maintain a copy of the entire database, but DB2 Extract Facility should be used to maintain a copy of only a subset of the database.

The contents of this subset vary, depending on the installation. Care needs to be taken when choosing the fields to be propagated, because this will affect the steps that are described below, in 8.8.1, “DB2 Extract Facility Checklist” on page 96.

The Automatic Log Save may not be maintaining a copy of the entire database. This depends upon whether you implement filtering, or whether you choose to propagate everything. The difference is in which side the reduction occurs on. With Automatic Log Save, it is on the receive side; with DB2, it is on the Information/Management side.

The DB2 administrator will need to be involved, because some of the steps involve the need for SYSADMIN authority in DB2.

It should be noted that there is no facility to do an initial load of the DB2 database.

## 8.8.1 DB2 Extract Facility Checklist

For the purposes of this checklist, fields have been chosen that represent a small subset of the Information/Management database, which are going to influence each of the steps shown below.

1. Create a DB2 database using the SPUFI commands shown in Figure 70. Also create a tablespace in this new database, as shown in Figure 71.

```
CREATE DATABASE INFODB2
   STOGROUP SYSSTGRP
   BUFFERPOOL BP0;

CREATE TABLESPACE INFOTB1
   IN INFODB2
   USING STOGROUP SYSSTGRP
             PRIQTY 20
             SECQTY 20
             ERASE NO
   LOCKSIZE PAGE
   BUFFERPOOL BP0
   PCTFREE 10
   CLOSE YES;

COMMIT ;
```

*Figure 70. Defining a DB2 Database*

2. Create a table in the tablespace to manage the synchronization of data. This is shown in Figure 71.

```
CREATE TABLE INFO1.SYNCH_TABLE
   (SYNCH_NUMBER INT)
   IN INFODB2.INFOTB1;

INSERT INTO INFO1.SYNCH_TABLE(SYNCH_NUMBER)
 VALUES(00000000);

COMMIT ;
```

*Figure 71. Defining a Synchronization Table*

3. Create a table in the tablespace that will be used to contain data from our Information/Management database. This is shown in Figure 72 on page 97. The fields in this figure need to match those that you choose to select when you run BLGUT8, to create your RDMT (Relational Data Mapping Table).

```
CREATE TABLE  INFO1.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)                ,
   PROBLEM_NUMBER   CHAR(08)                ,
   INITIAL_PRIORITY CHAR(02)                ,
   DESCRIPTION      CHAR(45)                ,
   PRIMARY KEY (RNID))
   IN INFODB2.INFOTB1;
CREATE UNIQUE INDEX INFO1.INFOIX1
   ON INFO1.PROBLEM_DATA(RNID);

COMMIT;
```

*Figure 72. Defining a Table to Contain Data*

4. Create a table in the tablespace to contain freeform text from our Information/Management database. This is shown in Figure 73.

```
CREATE TABLE  INFO1.PROBLEM_TEXT
  (RNID             CHAR(08)        NOT NULL,
   SEQ_NUM          CHAR(08)        NOT NULL,
   DESCRIPTION      CHAR(250)               ,
   PRIMARY KEY (RNID,SEQ_NUM))
   IN INFODB2.INFOTB1;
CREATE UNIQUE INDEX INFO1.INFOIX2
   ON INFO1.PROBLEM_DATA(RNID);

 COMMIT;
```

*Figure 73. Defining a Table to Hold Freeform Text*

5. BLMSTEP2

6. Create a GDG to contain the data that will be propagated to DB2. There is a sample job, BLGDG1, in the SBLMSAMP data set.

7. Run the BLGUT8 utility to create an RDMT. An example of this is shown in Figure 74 on page 98. Please note that an existing PIDT could be used, and SQLTAB and SQLCOL fields added, in which case any PIDT parms will be ignored. Conversely, if you are creating a PIDT, then the SQL parms will be ignored. Hence, you can maintain one set of source statements to cover the two situations.

```
//jobcard                                                               00001000
//*-------------------------------------------------------------------* 00020000
//STEP1    EXEC  PGM=BLGUT8,REGION=2048K                                00270000
//STEPLIB  DD DSN=your.SBLMMOD1,DISP=SHR                                00280000
//SYSPRINT DD SYSOUT=*                                                  00290000
//BLGDICT  DD DSN=your.DICT,DISP=SHR                                    00310000
//BLGPNLS  DD DSN=your.SBLMPNLS,DISP=SHR                                00320000
//BLGRFT   DD DSN=your.SBLMFMT,DISP=OLD                                 00330000
//SYSIN    DD *,DLM='%%'                                                00340000
                                                                        00591000
TABLE NAME(DB2XT1) USE(SQLMAP);                                         00592000
                                                                        00593000
FIELD PANEL(BLG00000) INDEX(S0032)        /*  PROBLEM RECORD TYPE  */   00594000
      RCDSWORD(Y);                                                      00595000
                                                                        00596000
FIELD PANEL(BLG6REQN) INDEX(S0B59)      /*  REPORTER NAME        */     00599000
      SQLTAB(PROBLEM_DATA) SQLCOL(REPORTER_NAME);                       00599100
                                                                        00599200
FIELD PANEL(BLG6PTYP) INDEX(S0C09)      /*  PROBLEM TYPE         */     00599300
      SQLTAB(PROBLEM_DATA) SQLCOL(PROBLEM_TYPE);                        00599400
                                                                        00599500
FIELD PANEL(BLG6RQDP) INDEX(S0B9B)      /*  REPORTER DEPT        */     00599600
      SQLTAB(PROBLEM_DATA) SQLCOL(REPORTER_DEPT);                       00599700
                                                                        00599800
FIELD PANEL(BLG6STAT) INDEX(S0BEE)      /*  REPORTER STATUS      */     00599900
      SQLTAB(PROBLEM_DATA) SQLCOL(REPORTER_STAT);                       00600000
                                                                        00610000
FIELD PANEL(BLG6PHON) INDEX(S0B2D)      /*  REPORTER PHONE       */     00620000
      SQLTAB(PROBLEM_DATA) SQLCOL(REPORTER_PHONE);                      00630000
                                                                        00640000
FIELD PANEL(BLG6URNO) INDEX(S0CCF)      /*  USER PROBLEM NUMBER  */     00650000
      SQLTAB(PROBLEM_DATA) SQLCOL(PROBLEM_NUMBER);                      00660000
                                                                        00670000
FIELD PANEL(BLG6PRII) INDEX(S0BE6)      /*  INITIAL PRIORITY     */     00680000
      SQLTAB(PROBLEM_DATA) SQLCOL(INITIAL_PRIORITY);                    00690000
                                                                        00700000
FIELD PANEL(BLG6DSAB) INDEX(S0E0F)      /*  DESCRIPTION          */     00710000
      SQLTAB(PROBLEM_DATA) SQLCOL(DESCRIPTION);                         00720000
                                                                        00721000
FIELD PANEL(BLG0B010) INDEX(S0E01)      /*  TEXT,DESCRIPTION     */     00722000
      SQLTAB(PROBLEM_TEXT) SQLCOL(TEXT) TEXT(Y);                        00723000
                                                                        00724000
                                                                        00730000
ETABLE;                                                                 00880000
%%                                                                      00890000
//                                                                      00900000
```

*Figure 74. Running BLGUT8 to Create an RDMT*

8. Update the LOGSAVE record on the send database.

   This record needs to be updated to provide the following information:

   - A data set name for the SQL input data that Automatic Log Save will create.
   - The names of the Relational Data mapping Table entries (RDMTs), created with BLGUT8.
   - The user ID that is authorized to the DB2 database.

9. Modify TSP BLGTDBX to specify the name of the JCL data set that contains the jobs outlined below.

10. Customize the following supplied jobs to suit your installation standards. In each case follow the instructions in the comments of each job.

    - BLMSDSD

      This job runs on the MVS image where the send database resides. It starts TSP BLGTDBXM in batch mode, and sends a status message to notify success or failure.

    - BLMSDCS

      This job runs on the MVS image where the send database resides. It copies the Automatic Log Save Send data set to the next generation of a GDG, and then deletes the Automatic Log Save Send data set.

    - BLMSSND

      This job which runs on the MVS image where the send database resides. It builds the job stream for BLMSRCV containing the SQL statements, then submits the job to the internal reader for routing to the system where DB2 resides.

    - BLMSL1

      This is a partial piece of JCL that is appended to BLMSRCV.

    - BLMSL2

      This is a partial piece of JCL that is appended to BLMSRCV.

    - BLMSRCV

      This job, which runs on the MVS image where the DB2 database resides, runs DSNTEP2 to process the SQL statements in batch, and sends a status message to notify success or failure.

### 8.8.2 Starting the DB2 Extract Facility

When the setup on the send database, and on the DB2 database, has been successfully completed, then the DB2 Extract Facility is started by submitting the batch job BLMSDSD.

### 8.8.3 Stopping the DB2 Extract Facility

The DB2 Extract Facility is stopped by changing the status of the LOGSAVE record from ENABLED to DISABLED. To achieve this, issue the following command from the Information/Management Primary Options Menu:

```
upd r logsave,4,9
```

If you issue this command a second time, the status will revert to ENABLED.

For further information about the DB2 Extract Facility, please refer to the *Program Administration Guide and Reference, SC34-4341*.

# Chapter 9. Maintenance

This chapter will cover recommended guidelines for maintenance purposes. The following areas will be covered:

- Control of Delta modules - these are modules that are required either for adding a new function, for example BLXCBIAS, which is the module for Multisystem Database Access, or for customization of BLGSESxx, the session parameters. These modules are installed by the user outside SMP/E, that is why they are referred to as Delta modules in this section.
- Housekeeping - Back-up and the reorganization of the Information/Management databases
- Data set analysis reports - Reports to provide information that will assist in tuning.

## 9.1 Control of Delta Modules

There are now a significant number of modules that can be generated after the SMP/E process has completed. There are three categories of modules:

1. Modules that have been created by customization jobs and link-edited into the SBLMMOD1 load library. These modules may need to be relinked after future software maintenance has been applied to the base code. Therefore, we recommend that the creation or updating of these modules is done in the form of SMP/E user modules. SMP/E will then ensure that when there is an update that directly or indirectly affects these modules, an updated copy of the module is placed in the load library. SMP/E can also be used to coordinate updates to these modules. The following modules are examples of this:

   - BLXCBIAS - The Multisystem Database Access module
   - BLMYSCSC - The Remote Environment Server (RES) support module

2. Modules that are liable to be modified on a regular basis, again link-edited into the SBLMMOD1 library. These are only going to be affected by maintenance applied to the macros used in the assembly process. However, it is good practice to make them SMP/E user modules too. The following modules are examples of this:

   - BLXVDEF - The VSAM buffers member
   - BLGSESxx - The session-parameter member

3. The last category covers a module that needs to be on a separate load library in a Multisystem Database Access single system environment. This is because this module contains an eye-catcher with the name of the BLX-SP subsystem. The module is BLXSSINM. This can′t be treated as an SMP/E user module, because there can be more than one version of this module, with the same member name, if Multisystem Database Access is implemented in a single system environment. Therefore, care needs to be taken to ensure that it does not accidentally get over-written or deleted.

An example of an SMP/E user module is shown in Figure 75 on page 102.

```
++ USERMOD (INFOMOD) .
++ VER(Z038) FMID(HOY6101) .
++ SRC (BLGSESIA) TXLIB(USERSRCE) DISTLIB(AUSERSRC).
++JCLIN .
//LKED      EXEC PGM=IEWL,PARM=′LET,LIST,XREF,NCAL′
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   UNIT=SYSDA,SPACE=(CYL,1)
//AUSERMOD DD   DISP=SHR,DSN=your.USERSRCE
//SYSLMOD  DD   DISP=SHR,DSN=your.SBLMMOD1
//SYSLIN   DD   *
  INCLUDE  AUSRMODS(BLGSESxx)
  ENTRY    BLGSESxx
  NAME     BLGSESxx(R)
/*
```

*Figure 75. An Example SMP/E User Module for a Session Parameter*

## 9.2 Housekeeping Activity

It is important that regular housekeeping jobs are scheduled. In the past, the following approach might have been considered to balance the availability of the Information/Management database against the benefits of running the housekeeping jobs.

- Nightly repro of the following key data sets to a GDG, probably on tape:
  - SDDS
  - SDIDS

  The dumping of SDIDS is optional as it can be rebuilt from the SDDS using the BLGUT1 utility.
- Nightly UT4 off-load of the SDLDS
- Weekly reorganization of the SDDS and SDIDS to remove CI and CA splits.

There is no reason why this approach could not continue to be used with Version 6 Release 1. However, the housekeeping suites will need to be rewritten, if Automatic Log Save is implemented, as this takes away the need for a nightly off-load of the SDLDS data set.

Consider simplifying the housekeeping jobs as follows

- Nightly repro of the following key data sets to a GDG, probably on tape:
  - SDDS
  - SDIDS
- Weekly reorganization of the SDDS and SDIDS to remove CI and CA splits or rebuilding the SDIDS with BLGUT1.

A factor that will influence the housekeeping jobs is availability. You cannot run the nightly repro of the SDDS and SDIDS without making these data sets unavailable to users, that is effectively closing down Information/Management.

Hence in an environment where you are maintaining 24-hour, seven day a week availability, we recommend that you consider taking the following approach to achieve high availability:

- Implement Automatic Log Save

- Do not run any daily backup jobs
- Run a weekly reorg of the SDDS and SDIDS

### 9.2.1 Weekly Housekeeping Procedure

The following approach should be considered for the weekly housekeeping jobs.

1. Stop the BLX1 server with the operator command `P BLX1`
2. OR free the SDDS, SDIDS and the SDLDS using the commands:

   ```
   F BLX1,FREE,FILE=MGMTSDDS

   F BLX1,FREE,FILE=MGMTINDX

   F BLX1,FREE,FILE=MGMTLOG
   ```

   The operand on the FILE keyword, for each of these commands, must match the label on the corresponding entry in the BLXVDEF definition.

   The last procedure is recommended if more than one SDDS and SDIDS use the same BLX-SP procedure.

3. Run one or more jobs to achieve the following. An example of the JCL to do this with a single SDDS is shown in Figure 76 on page 104.
   - Run an IDCAMS LISTC to list the attributes of the data set. Put the output in a data set for comparison purposes.
   - Run an IDCAMS repro offload for each SDDS data set
   - Run an IDCAMS Delete/Define for each SDDS data set
   - Run an IDCAMS repro reload for each SDDS data set
   - Run an IDCAMS LISTC to list the attributes of the data set. Put the output in a data set and compare it to the output above.

```
//jobcard
//*--------------------------------------------------------------
//*  List the SDDS file at file
//*--------------------------------------------------------------
//LISTC1   EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD DSN=listcat.output.dataset,DISP=SHR
//SYSIN    DD *,DLM='$$'
   LISTC ENT(your.SDDS) ALL
$$
//*--------------------------------------------------------------
//*  Repro  SDDS to a flat file
//*--------------------------------------------------------------
//REPSD2   EXEC PGM=IDCAMS,REGION=4096K,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SDDS     DD DSN=your.SDDS,DISP=SHR
//XSDDS    DD DSN=your.BACKUP.SDDS,DISP=SHR
//SYSIN    DD *,DLM='$$'
   REPRO INFILE(SDDS) OUTFILE(XSDDS)
$$
//*--------------------------------------------------------------
//*  Delete and redefine SDDS
//*--------------------------------------------------------------
//DELDEF3  EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *,DLM='$$'
    DELETE (your.SDDS) CLUSTER PURGE
    DEFINE -
      CLUSTER(NAME(your.SDDS)           -
         INDEXED  -
         KEYS(7 0)  -
         SHAREOPTIONS(1)  -
         VOLUMES(vvvvvv) -
         UNIQUE) -
      DATA(NAME(your.SDDS.DATA) -
         CYLINDERS(10 1) -
         CISZ(4096) -
         RECORDSIZE(4000 4089) -
         SPEED) -
      INDEX(NAME(your.SDDS.INDEX) -
         NOIMBED -
         NOREPLICATE )
$$
//*--------------------------------------------------------------
//*  Repro to the new SDDS file
//*--------------------------------------------------------------
//REPSD4   EXEC PGM=IDCAMS,REGION=4096K,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//XSDDS    DD DSN=your.BACKUP.SDDS,DISP=OLD
//SDDS     DD DSN=your.INFOA.SDDS,DISP=OLD
//SYSIN    DD *,DLM='$$'
  REPRO IFILE(XSDDS) OFILE(SDDS)
$$
//*--------------------------------------------------------------
//*  List the New loaded SDDS file
//*--------------------------------------------------------------
//LISTC5   EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD DSN=listcat.output.dataset,DISP=SHR
//SYSIN    DD *,DLM='$$'
   LISTC ENT(your.SDDS) ALL
$$
/*
```

*Figure 76. An Example Reorg Job*

4. Run a job to perform an IDCAMS delete and redefine of the SDIDS. An example if this is shown in Figure 77 on page 105.

```
//jobcard
//*--------------------------------------------------------------
//*  DELETE AND REDEFINE THE SDIDS DATA SET
//*--------------------------------------------------------------
//S010     EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *,DLM='$$'
    DELETE (your.SDIDS) CLUSTER PURGE
   DEFINE -
      CLUSTER(NAME(your.SDIDS)           -
         INDEXED  -
         KEYS(32 0)  -
         SHAREOPTIONS(1)  -
         VOLUMES(vvvvvv) -
         UNIQUE) -
      DATA(NAME(your.SDIDS.DATA) -
         CYLINDERS(1 1) -
         CISZ(2048) -
         RECORDSIZE(35 2041) -
         FREESPACE(20 20) -
         SPEED) -
      INDEX(NAME(your.SDIDS.INDEX) -
         NOIMBED -
         NOREPLICATE )
$$
/*
```

*Figure 77. An Example Delete/Define Job*

5. Start the BLX1 server with the operator command:

   `S BLX1`

6. Free the SDLDS with the following operator command:

   `F BLX1,FREE,FILE=MGMTLOG`

   The operand on the FILE keyword must match the label on the corresponding entry in the BLXVDEF definition.

   The above command ensures that the users who need the SDLDS to access Information/Management do not get access to the database. The session parameters used for initializing the SDIDS as indicated in Figure 78 on page 106 must not have the SDLDS allocated to it.

7. Run a job to rebuild the SDIDS using BLGUT1. An example job to do this is shown in Figure 78 on page 106.

```
//jobcard
//*-------------------------------------------------------------
//*  BLGUT1 TO REBUILD THE SDIDS, BASED ON
//*  THE CURRENT SDDS
//*-------------------------------------------------------------
//S010    EXEC PGM=BLGUT1,PARM=′SESS=xx,NAME=5′,REGION=2048K
//STEPLIB  DD   DSN=your.SBLMMOD1,DISP=SHR
//BLGSD    DD   DSN=your.SDDS,DISP=SHR
//BLGSI    DD   DSN=your.SDIDS,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(10))
```

*Figure 78. An Example BLGUT1 Job*

8. Reallocate the SDLDS with the operator command:

   ```
   F BLX1,REALLOC,FILE=SDLDS
   ```

## 9.3 Data Set Analysis Reports

An extra job at the end of the housekeeping suite to run the following utilities will help you maintain an understanding of the attributes and contents of your VSAM data sets. It will also provide information to influence tuning, particularly in the area of LSR buffers.

**BLGUT20** This utility analyzes the contents of the SDDS data set. The output of BLGUT20 can be used to estimate the best control interval (CI) size, average record length, and maximum record length for the data set. The aim is to minimize the number of SDDS records that do not fit into a single CI without having a larger CI size than needed.

Figure 80 on page 108 and Figure 81 on page 108 gives an example of the summary messages and analysis from this utility.

**BLGUT21** This utility analyzes the contents of the SDIDS data set. The output of this utility can be used to estimate the best control interval (CI) size, average record length, and maximum record length for the data set. If you decide to change any of these values, then you will need to redefine the SDIDS with IDCAMS, and run BLGUT1 to rebuild its contents.

Figure 82 on page 108 and Figure 83 on page 109 gives an example of the summary messages and analysis from this utility.

**BLGUT22** This utility analyzes the contents of a panel′s data set. The output of this utility can be used to estimate the best control interval (CI) size, average record length, and maximum record length for the data set. The aim is to choose a CI size that allows most panels to fit into a single CI. If you decide to change any of these values, then you will need to:

1. Off-load the existing panel contents if you are analyzing or redefining a local panels data set.
2. Redefine the data set with IDCAMS
3. Run BLGUT6 to load the panels into it.

Figure 84 on page 109 and Figure 85 on page 110 gives an example of the summary messages and analysis from this utility.

The job shown in Figure 79 on page 107 gives an example of how to run these three utilities.

```
//jobcard
//*-------------------------------------------------------------
//*
//* BLGUT20
//*
//* DD NAME BLGSDDS IS THE NAME OF THE SDDS FROM WHICH TO WORK.
//* DD NAME ANALYZE IS THE NAME OF THE DATA SET TO CONTAIN THE OUTPUT.
//*
//* THE RECS PARAMETER FOR PROGRAM BLGUT20 REFERS TO THE NUMBER OF
//* LOGICAL RECORDS TO BE PROCESSED BETWEEN EACH RELEASE OF THE SDDS
//* DATA SET.  IN THIS EXAMPLE THE NUMBER IS SET TO 50.
//*-------------------------------------------------------------
//S010     EXEC PGM=BLGUT20,PARM=′RECS=50′,REGION=2048K
//STEPLIB  DD   DSN=BLM610.BASE.SBLMMOD1,DISP=SHR
//BLGSDDS  DD   DSN=BLM610.INFOA.SDDS,DISP=SHR
//ANALYZE  DD   DSN=BLM610.INFOA.SDDS.ANALYZE,DISP=OLD
//SYSPRINT DD   SYSOUT=*
/*
//*-------------------------------------------------------------
//*
//* BLGUT21
//*
//* DD NAME BLGSDIDS IS THE NAME OF THE SDIDS FROM WHICH TO WORK.
//* DD NAME ANALYZE IS THE NAME OF THE DATA SET TO CONTAIN THE OUTPUT.
//*
//* THE RECS PARAMETER FOR PROGRAM BLGUT21 REFERS TO THE NUMBER OF
//* LOGICAL RECORDS TO BE PROCESSED BETWEEN EACH RELEASE OF THE SDIDS
//* DATA SET.  IN THIS EXAMPLE THE NUMBER IS SET TO 50.  IN ADDITION,
//* SINCE THE NOCNT KEYWORD IS NOT SPECIFIED IN THE PARM FIELD, THE
//* BIT COUNT WILL BE GENERATED.
//*-------------------------------------------------------------
//S020     EXEC PGM=BLGUT21,PARM=′RECS=50′,REGION=2048K
//STEPLIB  DD   DSN=BLM610.BASE.SBLMMOD1,DISP=SHR
//BLGSDIDS DD   DSN=BLM610.INFOA.SDIDS,DISP=SHR
//ANALYZE  DD   DSN=BLM610.INFOA.SDIDS.ANALYZE,DISP=OLD
//SYSPRINT DD   SYSOUT=*
/*
//*-------------------------------------------------------------
//*
//* BLGUT21
//*
//* DD NAME BLGPANS IS THE NAME OF THE PANEL DATA SET USED AS INPUT.
//* DD NAME ANALYZE IS THE NAME OF THE DATA SET TO CONTAIN THE OUTPUT.
//*
//* THE RECS PARAMETER FOR PROGRAM BLGUT22 REFERS TO THE NUMBER OF
//* PANELS TO BE PROCESSED BETWEEN EACH RELEASE OF THE PANEL
//* DATA SET.  IN THIS EXAMPLE THE NUMBER IS SET TO 50.
//*-------------------------------------------------------------
//S030     EXEC PGM=BLGUT22,PARM=′RECS=50′,REGION=2048K
//STEPLIB  DD   DSN=BLM610.BASE.SBLMMOD1,DISP=SHR
//BLGPANS  DD   DSN=BLM610.INFOA.IBMPNLS,DISP=SHR
//ANALYZE  DD   DSN=BLM610.INFOA.PANEL.ANALYZE,DISP=OLD
//SYSPRINT DD   SYSOUT=*
```

*Figure 79. Running the Statistic Utilities*

```
SDDS DATA SET ANALYZED: 'BLM610.INFOA.SDDS'   TIME: 15:07:14 DATE: 05/05/94
CI SIZE: 4096     KEY LENGTH: 7     MAXIMUM RECORD LENGTH: 4089
TOTAL DATA SET SIZE: 325134
    TOTAL SDDS STRUCTURED DATA SIZE: 281629
    TOTAL SDDS STRUCTURED TEXT SIZE: 3018
    TOTAL SDDS STRUCTURED HISTORY SIZE: 40487
LOGICAL RECORDS PROCESSED: 254
PHYSICAL RECORDS PROCESSED: 255
MINIMUM SDDS LOGICAL RECORD SIZE: 1074
AVERAGE SDDS LOGICAL RECORD SIZE: 1280
MAXIMUM SDDS LOGICAL RECORD SIZE: 5258
SDDS RECORDS REQUIRING A  512 CI SIZE: 0
SDDS RECORDS REQUIRING A 1024 CI SIZE: 0
SDDS RECORDS REQUIRING A 2048 CI SIZE: 248
SDDS RECORDS REQUIRING A 3072 CI SIZE: 5
SDDS RECORDS REQUIRING A 4096 CI SIZE: 0
SDDS RECORDS REQUIRING A 6144 CI SIZE: 1
SDDS RECORDS REQUIRING A 8192 CI SIZE: 0
SDDS RECORDS REQUIRING A GREATER THAN 8192 CI SIZE: 0
SDDS RECORDS REQUIRING MULTIPLE CONTROL INTERVALS: 1
SDDS ANALYZE UTILITY COMPLETE
```

*Figure 80. Example BLGUT20 Summary Messages*

```
 DATE: 04/29/94   TIME: 17:02:33
 RECORD     KEY(1:5)    PHYSICAL
 LENGTH       SEQ#       RECORDS
--------  ------------- --------
00005258  X'0000000001' 00000002
              .
              .
              .
00001161  X'00000000FE' 00000001

(There will be one line in this report for each logical
record in the SDDS data set)
```

*Figure 81. Example BLGUT20 Analysis*

```
SDIDS DATA SET ANALYZED: 'BLM610.INFOA.SDIDS'   TIME: 15:07:22 DATE: 05/05/94
CI SIZE: 2048     KEY LENGTH: 32     MAXIMUM RECORD LENGTH: 2041
CURRENT NUMBER OF DATABASE RECORDS: 254
MAXIMUM NUMBER OF DATABASE RECORDS ALLOWED: 16008
TOTAL DATA SET SIZE: 30161
SDIDS RECORDS PROCESSED: 661
MINIMUM SDIDS RECORD SIZE: 41
AVERAGE SDIDS RECORD SIZE: 46
MAXIMUM SDIDS RECORD SIZE: 72
SDIDS SPANNED RECORDS REQUIRING MULTIPLE CONTROL INTERVALS: 0
SDIDS ANALYZE UTILITY COMPLETE
```

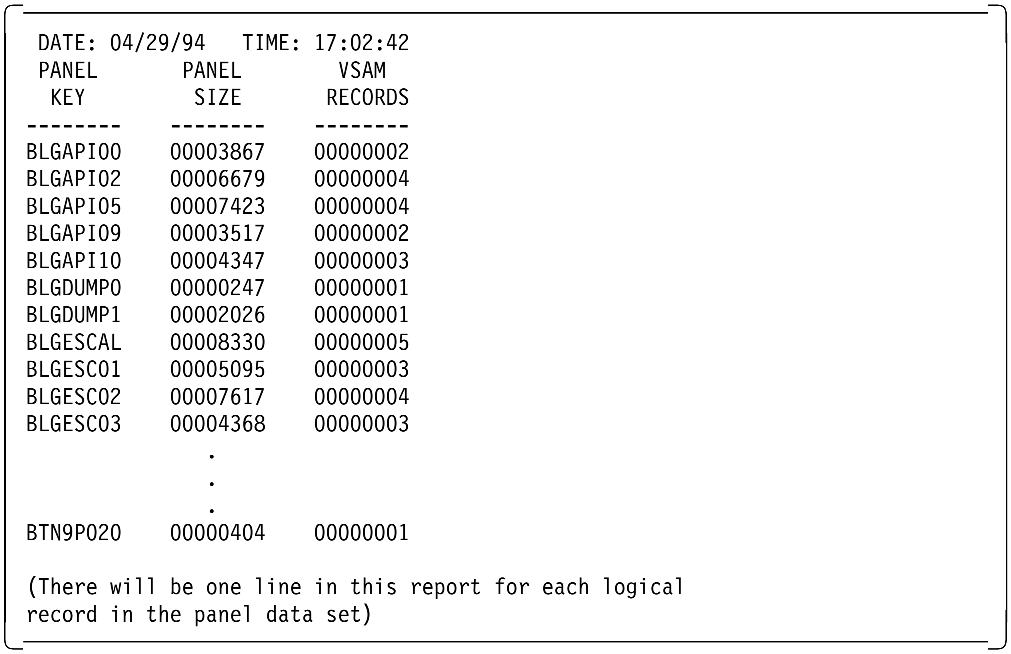
*Figure 82. Example BLGUT21 Summary Messages*

```
 DATE: 04/29/94   TIME: 17:02:38
 RECORD        SDIDS
 LENGTH         KEY

00000051  #01
00000041  .//S/MSI
00000044  .//S/TRSM
00000044  .//S/TRSP
00000072  .//S/TSI
00000072  .//S/TXS
00000044  .LASTENTRYNUMBER
00000072  .MASTER_BIT_LIST
00000044  'IMDIAADDO
            .
            .
            .
00000044  92008

(There will be one line in this report for each logical
record in the SDIDS data set)
```

*Figure 83. Example BLGUT21 Analysis*

```
 PANEL DATA SET ANALYZED: 'BLM610.INFOA.IBMPNLS'   TIME: 15:07:26 DATE:05/05/94
 CI SIZE: 4096     KEY LENGTH: 10     MAXIMUM RECORD LENGTH: 2041
 TOTAL DATA SET SIZE: 5377159
 LOGICAL RECORDS PROCESSED: 6017
 PHYSICAL RECORDS PROCESSED: 6451
 MINIMUM PANEL SIZE: 204
 AVERAGE PANEL SIZE: 894
 MAXIMUM PANEL SIZE: 17868
 PANELS REQUIRING A  512 CI SIZE: 1900
 PANELS REQUIRING A 1024 CI SIZE: 2570
 PANELS REQUIRING A 2048 CI SIZE: 1239
 PANELS REQUIRING A 3072 CI SIZE: 198
 PANELS REQUIRING A 4096 CI SIZE: 48
 PANELS REQUIRING A 6144 CI SIZE: 31
 PANELS REQUIRING A 8192 CI SIZE: 9
 PANELS REQUIRING A GREATER THAN 8192 CI SIZE: 22
 PANELS REQUIRING MULTIPLE CONTROL INTERVALS: 62
 PANEL ANALYZE UTILITY COMPLETE
```

*Figure 84. Example BLGUT22 Summary Messages*



```
 DATE: 04/29/94   TIME: 17:02:42
 PANEL       PANEL        VSAM
  KEY         SIZE       RECORDS
--------    --------    --------
BLGAPI00    00003867    00000002
BLGAPI02    00006679    00000004
BLGAPI05    00007423    00000004
BLGAPI09    00003517    00000002
BLGAPI10    00004347    00000003
BLGDUMP0    00000247    00000001
BLGDUMP1    00002026    00000001
BLGESCAL    00008330    00000005
BLGESC01    00005095    00000003
BLGESC02    00007617    00000004
BLGESC03    00004368    00000003
                .
                .
                .
BTN9P020    00000404    00000001

(There will be one line in this report for each logical
record in the panel data set)
```

*Figure 85. Example BLGUT22 Analysis*

For further information about the Information/Management utilities discussed here, please refer to the *Operation and Maintenance Reference Version 6 Release 1 for MVS/ESA,SC34-4480*.

## 9.4 Security Considerations

Information/Management uses two RACF* classes to define access to VSAM data sets. The authorities controlled by these classes are in addition to the RACF profiles protecting individual data sets. The classes are:

- INFOMAN - To define access to a single data set
- GINFOMAN - To define access to a group of VSAM data sets

### 9.4.1 Security Check List

The following requirements are mandatory when using RACF to control access to all VSAM data sets:

1. Protect all Information/Management data sets with a universal access of NONE. Beyond that, explicitly permit those people who need higher access. Information/Management System Administrators will require UPDATE access to all resources and data sets.
2. Define a resource profile in the INFOMAN RACF class for each VSAM data set, and maintain an access list for each of these resources, rather than permitting users to the VSAM data sets directly. Users will not write to the VSAM data sets because all I/O is serviced by the BLX-SP task. The BLX-SP queries the INFOMAN class to determine whether a user is authorized to access a given data set.
3. Create a RACF started task table entry for each BLX-SP. Associate a RACF user ID with each BLX-SP.
4. Give RACF user ID associated with the BLX-SP CONTROL access to all Information/Management data sets.

5. The GINFOMAN class allows multiple profiles, so you can have one access list controlling several data sets that have similar security requirements.

### 9.4.2 How to Create RACF Resource Definitions

You will use four commands to protect your resources in the classes INFOMAN and GINFOMAN. Please note that these commands must be executed by someone with group special, or with CLAUTH authority in INFOMAN and GINFOMAN. We recommend that the RACF administrator delegate the CLAUTH authority for these two classes to the Information/Management System Administrator.

The commands are:

1. RDEFINE - RDEF for short

   This command adds a resource to a RACF class. For example, if we wish to define a resource for a test database, with data sets INFO6.TEST.SDDS, INFO6.TEST.SDIDS, and INFO6.TEST.SDLDS, to be owned by a RACF group INFOADM, then we do this with the following command:

   ```
   RDEF GINFOMAN TESTDB ADDMEM(INFO6.TEST.SDDS INFO6.TEST.SDIDS
   INFO6.TEST.SDLDS) OWNER(INFOADM) UACC(NONE)
   ```

2. PERMIT - PE for short

   This command allows you to permit user IDs or groups to the protected resources. For example, if we want user ID SERGIO to be allowed to access our test SDDS, then we issue the following command:

   ```
   PE TESTDB CLASS(GINFOMAN) ID(SERGIO) ACC(UPDATE) UACC(NONE)
   ```

3. RALTER - RALT for short

   This command modifies a definition in a resource class. For example, if we changed the name of our test SDDS, then we could change our profile, with the following commands:

   ```
   RALT GINFOMAN TESTDB DELMEM(INFO6.TEST.SDDS INFO6.TEST.SDIDS
   INFO6.TEST.SDLDS)
   ```

   Please note that the access list would have to be rebuilt, and therefore should be listed before executing the RALT.

   ```
   RALT GINFOMAN TESTDB ADDMEM(INFO6.DEVT.SDDS INFO6.DEVT.SDIDS
   INFO6.DEVT.SDLDS) OWNER(INFOADM) UACC(NONE)
   ```

4. RDELETE - RDEL for short

   This command deletes a definition in a resource class. For example, if we were no longer going to run our test database on this machine, then we could delete our resource profile, with the following command:

   ```
   RDEL GINFOMAN TESTDB
   ```

To activate any changes to the above resources will need one or both of the following commands to be issued:

- SETROPTS RACLIST(INFOMAN)
- SETROPTS RACLIST(GINFOMAN)

# Index

# ITSO Technical Bulletin Evaluation RED000

**Installing Information/Management Version 6 Release 1**
**for Client/Server**

**Publication No. GG24-4323-00**

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

**Please rate on a scale of 1 to 5 the subjects below.**
**(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)**

| | | | |
|---|---|---|---|
| **Overall Satisfaction** | ____ | | |
| Organization of the book | ____ | Grammar/punctuation/spelling | ____ |
| Accuracy of the information | ____ | Ease of reading and understanding | ____ |
| Relevance of the information | ____ | Ease of finding information | ____ |
| Completeness of the information | ____ | Level of technical detail | ____ |
| Value of illustrations | ____ | Print quality | ____ |

**Please answer the following questions:**

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

________________________________________

________________________________________

What other topics would you like to see in this Bulletin?

________________________________________

________________________________________

What other Technical Bulletins would you like to see published?

________________________________________

**Comments/Suggestions: ( THANK YOU FOR YOUR FEEDBACK! )**

________________________ ________________________
Name Address

________________________ ________________________
Company or Organization

________________________ ________________________
Phone No.

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department 985, Building 657
P.O. BOX 12195
RESEARCH TRIANGLE PARK NC
USA 27709-2195

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.

GG24-4323-00