INTERCOMM

MULTIREGION SUPPORT FACILITY

ISOGON
330 Seventh Avenue · New York, New York 10001

LICENSE: INTERCOMM TELEPROCESSING MONITOR

Multiregion Support Facility

Publishing History

Publication	Date	Remarks
First Edition	December 1974	This is a new manual, corresponding to Intercomm Release 6.2.
IPN 099	April 1976	Release 7.0 updates.
IPN 114	December 1976	General updates.
Second Printing	January 1977	Incorporating IPN 114.
IPN 123	April 1977	RAP security and general updates.
Second Edition	July 1979	General updates and revisions, corresponding to Intercomm Release 8.0.
Third Edition	March 1981	Revisions and updates.

PREFACE

Intercomm is a state-of-the-art teleprocessing monitor system operating under the control of IBM 360/370 operating systems (MFT, MVT, VS). Intercomm monitors the transmission of messages from terminals, concurrent message processing, centralized access to I/O files, and the routine utility operations of editing input messages and formatting output messages, as required.

Multiregion Support Facility documents the use of Multiregion Support (MRS), an extended capability of Intercomm that allows the support of application programs in more than one system region at a time.

The reader is assumed to be familiar with the installation and operation of Intercomm. The Operating Reference Manual and Basic System Macros are to be referenced in conjunction with this publication.

A Users Review Form is included at the back of this manual. We welcome recommendations, suggestions and reactions to this or any Intercomm publication.

This edition of the Multiregion Support Facility corresponds to Intercomm Release 8.0.

INTERCOMM PUBLICATIONS

GENERAL INFORMATION MANUALS

Concepts and Facilities

Planning Guide

APPLICATION PROGRAMMERS MANUALS

Assembler Language Programmers Guide

COBOL Programmers Guide

PL/1 Programmers Guide

SYSTEM PROGRAMMERS MANUALS

Basic System Macros

BTAM Terminal Support Guide

Installation Guide

Messages and Codes

Operating Reference Manual

System Control Commands

CUSTOMER INFORMATION MANUALS

Customer Education Course Catalog

Technical Information Bulletins

User Contributed Program Description

FEATURE IMPLEMENTATION MANUALS

Autogen Facility

ASMF Users Guide

DBMS Users Guide

Data Entry Installation Guide

Data Entry Terminal Operators Guide

Dynamic Data Queuing Facility

Dynamic File Allocation

Extended Security System

File Recovery Users Guide

Generalized Front End Facility

Message Mapping Utilities

Multiregion Support Facility

Page Facility

Store/Fetch Facility

SNA Terminal Support Guide

Table Facility

TCAM Support Users Guide

Utilities Users Guide

EXTERNAL FEATURES MANUALS

SNA LU6.2 Support Guide

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The Intercomm Multiregion Support Facility (MRS) permits groups of application subsystems to operate in specific OS/VS partitions or regions called satellite regions. Application subsystems in satellite regions execute independently from each other and from the control region. The Intercomm Front End is contained in the control region, which may also contain application subsystems. Application programs executing in batch regions or partitions may send messages directly to the control region or to other satellite regions via the control region. Abnormal termination of a satellite region does not affect the operation of the control region. Thus, the terminal operator is protected from application program failure (beyond normal recovery from program checks).

Satellite regions are also protected from each other because files are defined to each region via JCL; file access should be restricted by the user to one region only, affording complete file protection and region-oriented exclusive control for update processing. Message restart and the recovery of files following abnormal termination in one region does not affect other regions unless files are shared across regions. Updating of a particular VSAM file must be restricted to one region (see IBM share options restrictions). Inquiry against a VSAM file may be performed in another region if the share options are used. The mutual isolation of regions, application subsystems, and application files also provides additional security under Intercomm.

Separate and multiple satellite regions also effect decentralization of implementation, control and maintenance. The responsibility of a single teleprocessing control group to monitor the overall system in an installation is lessened as each satellite region has its own control environment. Each satellite region may belong to a functional group within an organization. Each group independently implements, tests and maintains its own satellite region. This concept is consistent with the application project team concept that allows testing of new applications without affecting existing applications.

The functional aspects of an Intercomm multiregion environment are directed by the control region, which processes all terminal and interregion message traffic, that is, the routing of messages to and from satellite regions. This region knows the location of every subsystem in the multiregion environment. The satellite regions that are initiated, the actions taken if a region is inactive, queuing requirements for every region, logging requirements, and so forth, are defined by the user via the Region Descriptor Table (RDT). Within the control region, an Intercomm system program, the Region Queue Manager, processes the queuing and dequeuing of messages destined for regions. Whenever a region is inactive, appropriate actions on a region and subsystem basis are also prescribed for the Region Queue Manager.

Satellite regions consist of application subsystems and an Intercomm Back End, (that is, the Subsystem Controller, File Handler, and all required service routines). Satellite regions directly communicate only with the control region, and indirectly with other satellite regions via the control region. Subsystems which conform to standard Intercomm coding conventions need not be modified to execute in either a satellite region or the control region. Output messages produced by subsystems in satellite regions that are not destined for other subsystems in the same region are routed by Message Collection (the queuing module) or FESEND to the Multiregion Output Subsystem (an Intercomm-supplied program) for transfer to the control region. Such messages may then be queued for Front End transmission, routed to other satellite regions, or processed by subsystems in the control region (such as the Output Utility).

Satellite regions may communicate with each other, but only through the control region, providing independence and flexibility in the scheduling and composition of the satellite regions. Every satellite region that is initiated is given its own set of ECB channels. Communication is accomplished by the use of these channels. Multiple batch regions, however, all share the same set of ECB channels. For interregion message transfer, the sending region posts an ECB with the address of the message to be transferred; the receiving region, alerted by the posting of the ECB, then copies the message into its dynamic storage and notifies the sending region of its acceptance of the message (by posting an acknowledgement ECB).

The control region is notified by all satellite regions when they start up or close down. The control region notifies all satellite regions of its termination, whether normal or abnormal. Abnormal region termination is provided for by STAEEXIT system routines in Intercomm. If a region abends without giving notification through STAEEXIT, the abnormally terminated region is still detected, since each region monitors the region or regions with which it is communicating.

The following characterize the Intercomm Multiregion Support Facility:

- Terminal I/O is centralized in one region.
- Applications are decentralized into one or more regions.
- System integrity, security and high reliability are provided by separating and shielding distinct applications from each other.
- Monitor integrity is enhanced because terminals are less likely to be affected by application failure when the Front End is separated from applications residing in satellite regions.
- All regions may be started and stopped in any order.

- Particular subsystem activity (input message queuing) may be started and stopped.

- No currently operational subsystem need be modified to run in the multiregion environment.

- Absolute file protection is attained when files are associated with only one region by JCL; file access is absolutely restricted to the set of programs in that region. Also, since Restart/Recovery is by region, the recovery of files in one region is independent of subsystem execution in other regions in the Intercomm File Recovery scheme.

- Separate logging is provided in each region. Optionally, all logging may be to a single INTERLOG data set belonging to the control region, however, Restart/Recovery cannot be used.

- Automatic and independent handling of region abends is provided.

- Concurrent testing of a new application is fail-safe with MRS. A test region can be used to implement new programs without jeopardizing the production system.

- The batch interface allows batch programs to direct messages to the on-line system via a service routine provided with MRS.

- Messages to a subsystem in a particular region that is inactive may be flushed, queued for later processing or sent to an alternate active region.

- Page fault overlap is automatically accomplished for all VS systems.

Region Associated Processing (RAP), an optional facility of MRS, is a security feature that allows the user to lock a terminal to one specific satellite region. The lock prevents the terminal operator from accessing programs or data sets other than those available in the specified satellite region or the control region. Although a terminal may be locked to a verb associated with a production region, that terminal may be dynamically locked to a test region for testing modifications to an existing subsystem associated with the same verb, or unrelated existing or new subsystems (after being unlocked from the production verb). Association of a terminal to a particular region is controlled by user-assigned passwords. Under RAP, message traffic between satellite regions is not allowed.

Chapter 2

OPERATING CONCEPTS

All message flow in the multiregion environment is directed by the control region:

- Input from terminals may be directed to a satellite region for processing.
- Input from terminals may be directed to a control region subsystem, such as the Change/Display Utility or a user subsystem.
- Input from a satellite region may be directed to the control region or another satellite region for processing.
- Input from a batch region may be directed to the control region or a satellite region subsystem.

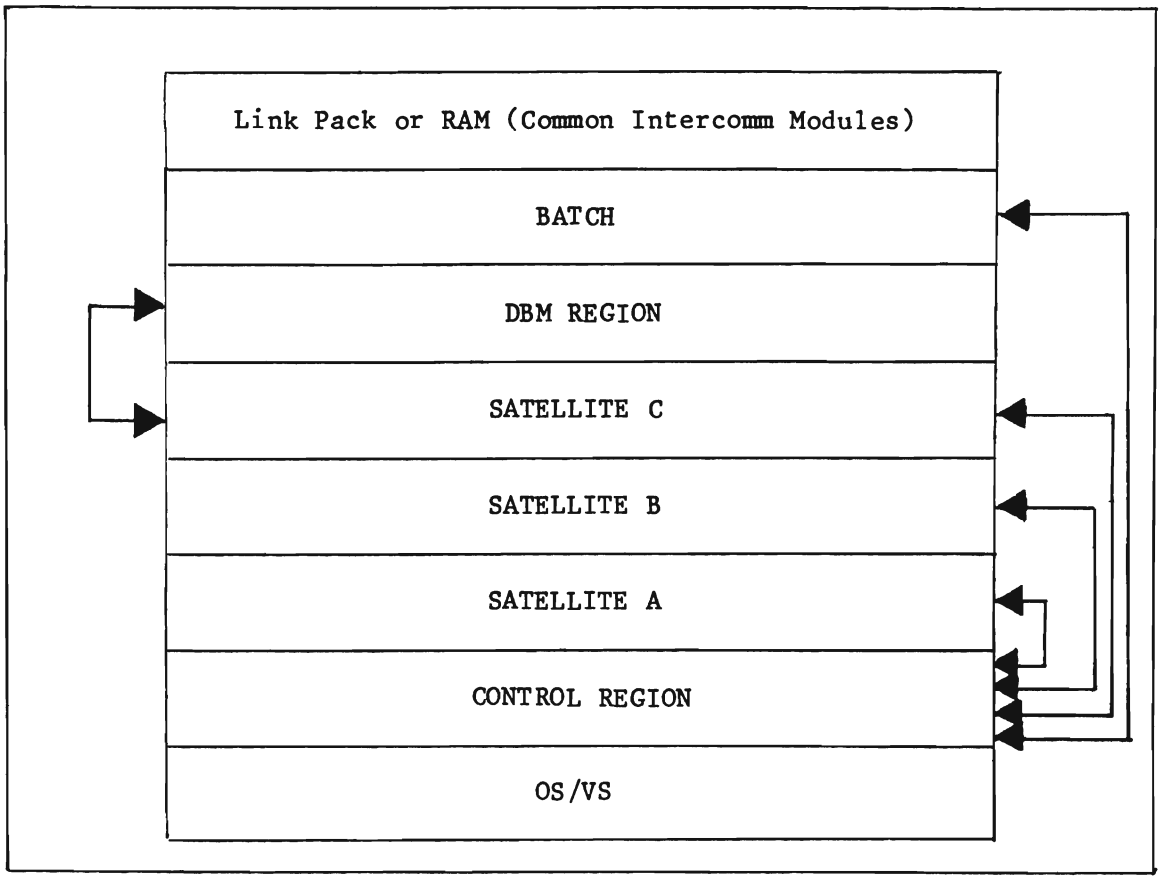

Figure 1. The Multiregion Environment

An output message produced by an application subsystem identifies the next subsystem to process a message via the receiving subsystem code in the standard Intercomm message header. In the satellite region, the Queue Management routine (message collection) routes any message not destined for a subsystem within that region (that is, not in the Subsystem Control Table) to the Multiregion Output Subsystem, which transfers the message to the control region. An output message destined for a terminal is queued for the control region via a subsystem call to FESEND (FESENDC). Figures 1, 2 and 3 illustrate interregion traffic flow.

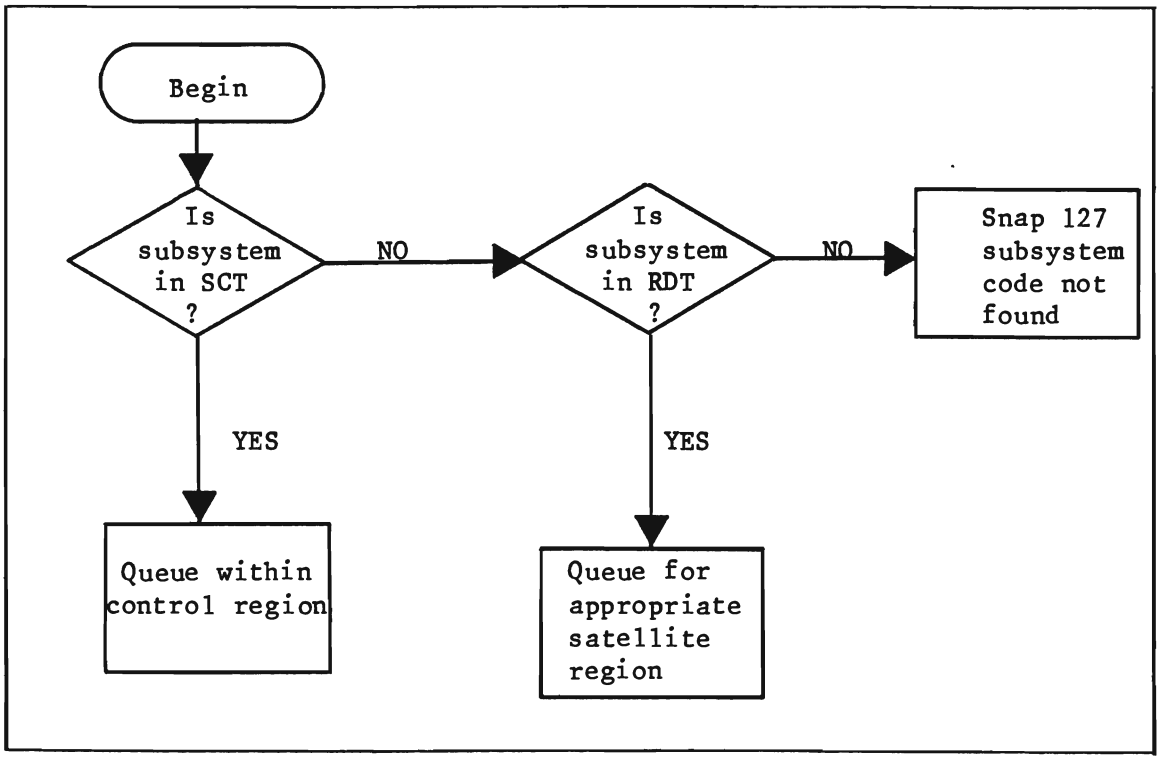

Figure 2. Queuing Logic in the Control Region

2.1 MULTIREGION QUEUES

There are two types of message queues used by MRS in the control region: interregion queues and subsystem "hold" queues. These are distinct from terminal queues for messages destined for terminals and subsystem queues for messages that will be executed by subsystems within the control region.

These MRS message queues are used for messages awaiting interregion transfer. The queues are defined on a region basis. Interregion queues may be core and/or disk queues. Subsystem (hold) queues are disk queues only, independent of the normal control region queues. Figure 4 illustrates the level of queues in the multiregion environment.

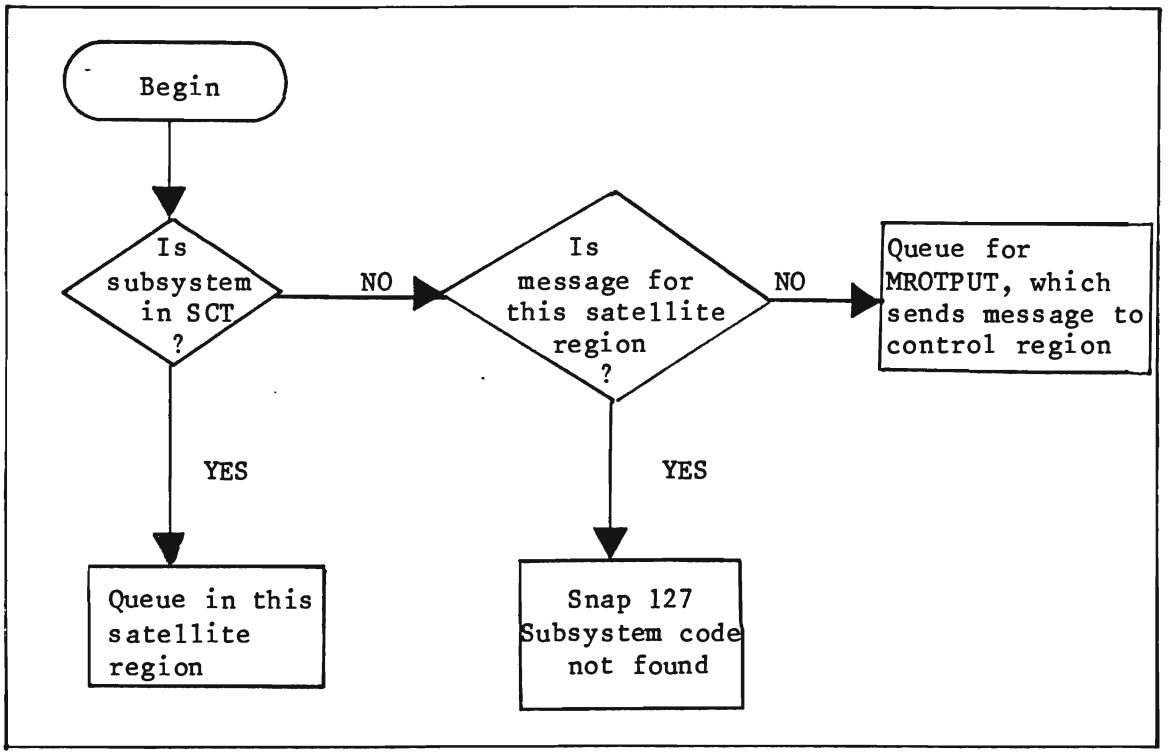

Figure 3. Queuing Logic for Satellite Regions

The subsystem hold queues and the disk components of interregion queues are maintained on dynamic data queues (DDQs) using the Dynamic Data Queuing Facility. These DDQs are dynamically constructed when any interregion message traffic exists that must be queued and are deleted when empty.

The Multiregion Queue Manager, a module in the control region, maintains the interregion core and/or disk queues. Messages are processed on a first in/first out basis, with the disk queue acting as overflow space for the core queue should both core and disk queues be defined for interregion processing.

Messages destined for an inactive region may be:

- Routed to an alternate (active) region.
- Rejected with an appropriate message to the terminal operator.
- Queued on a subsystem hold queue until the satellite region becomes active. As noted, subsystem hold queues are maintained on disk only.
- Message disposition is specified by the ALT and IFDOWN parameters of the SUBSYS macro in the Region Descriptor Table.

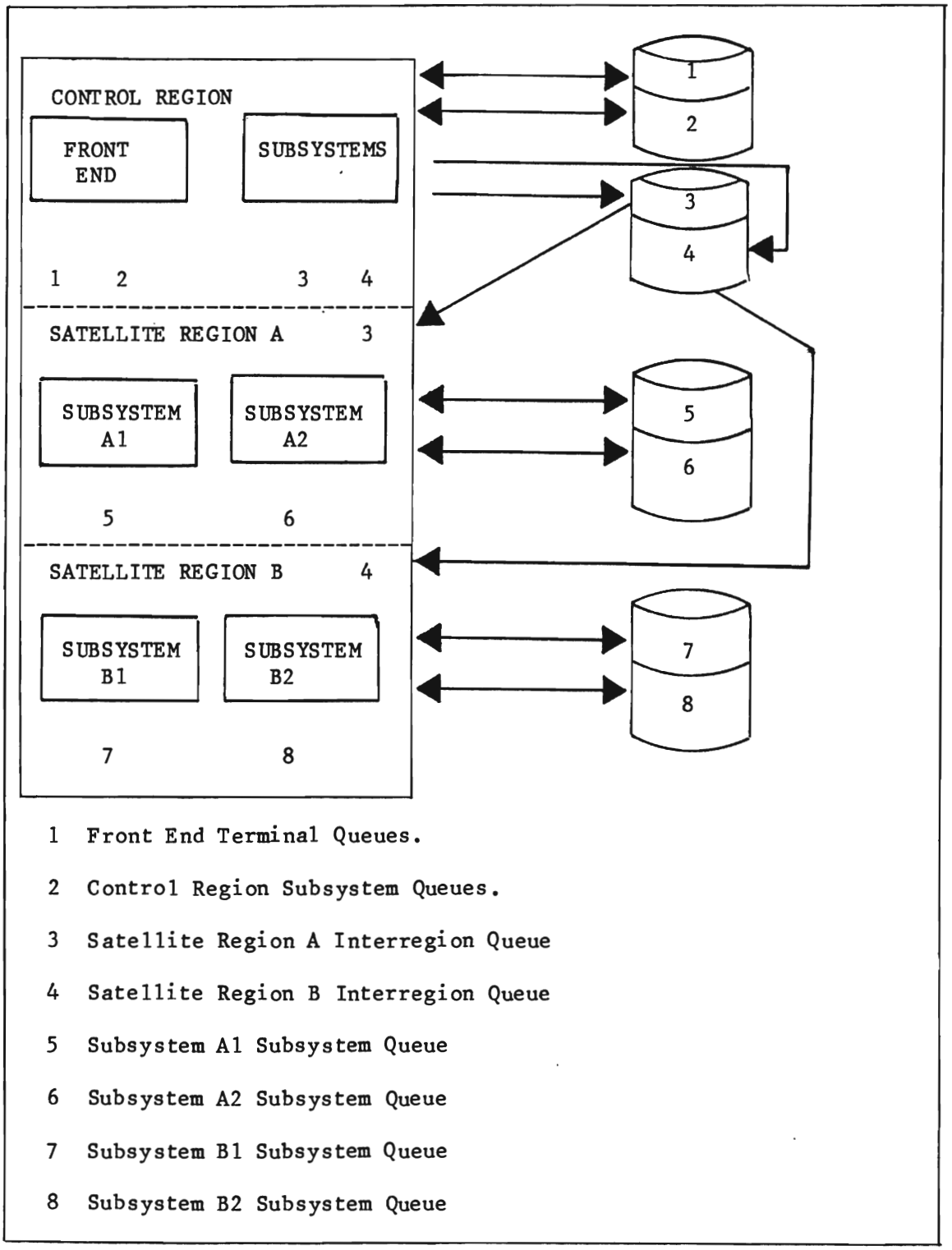

Figure 4. Queues in the Multiregion Environment

The following messages are returned to terminals entering messages to inactive regions (either not initiated yet or terminated normally or abnormally) or to regions (or subsystems) stopped via operator command:

REGION xxxxxxxx IS INACTIVE. YOUR MESSAGE WAS FLUSHED. REGION xxxxxxxx IS INACTIVE. YOUR MESSAGE WAS QUEUED. xxxxxxxx is the region identifier. The message disposition depends on the coding of the RDT SUBSYS macro IFDOWN parameter.
PROGRAM TEMPORARILY STOPPED. YOUR MESSAGE WAS FLUSHED. Subsystem (or region) was stopped by the MRS command subsystem processing of a STOP command from a terminal operator.

2.2 MULTIREGION TABLES

The multiregion environment is described by the Multiregion Communications Table, the Region Descriptor Table, and the System Parameter Area. Figure 5 illustrates the tables and their relationship. Coding specifications for the required macros are given in Appendix A.

2.2.1 Multiregion Communications Table (MCT)

Resident in the OS/VS Link Pack or RAM area, this table defines all satellite regions by an eight-character identifier and defines any batch regions used to send messages to Intercomm. The control region is not defined here. The MCT provides space for interregion communications areas (ECB channels). For interregion message transfer, the sending region posts an ECB with the address of the message to be transferred; the receiving region, alerted by the posting of the ECB, then copies the message into its dynamic storage and notifies the sending region of its acceptance of the message by posting an acknowledgment ECB. In an MVS environment, the MCT must reside in the Fixed Link Pack Area.

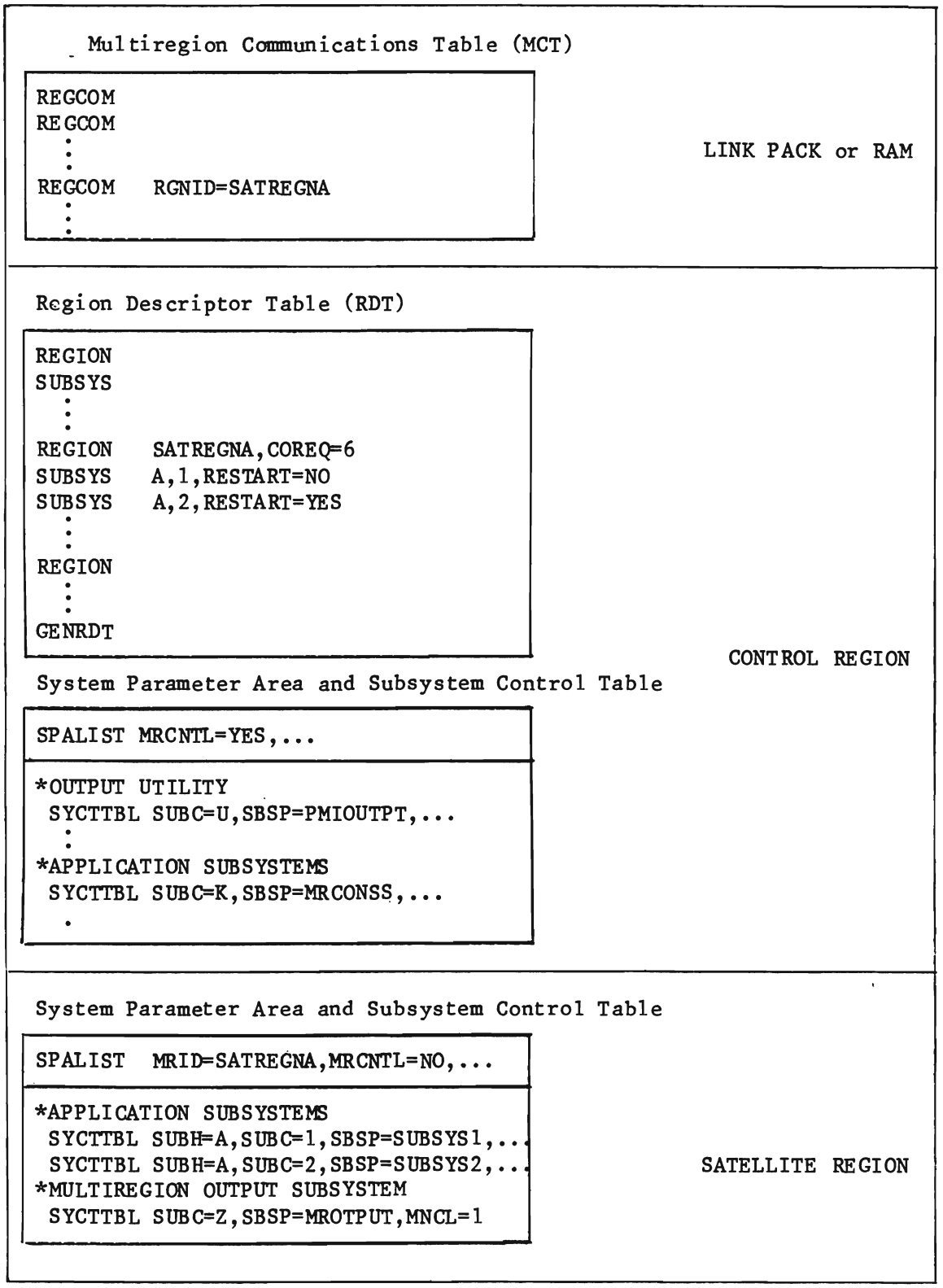

Figure 5. Multiregion Tables
(Macro coding illustrates key parameters only)

2.2.2 Region Descriptor Table (RDT)

Resident in the control region, this table defines all satellite regions and their associated subsystems. For each subsystem, the RDT specifies the action to take for an input message if a region is inactive, and queuing and logging requirements. This table is loaded dynamically at system startup, based upon operator specification of the member name. Thus, several different multiregion configurations may be defined by different RDTs.

2.2.3 System Parameter Area (SPA)

Resident in each region, the System Parameter Area is a unique table with operands specifying the region type and region identifier. This table is followed by the standard Subsystem Control Table (SCT) entries for each subsystem within the region.

2.3 SYSTEM CONTROL AND SYSTEM RESTART

The control region and satellite regions operate independently as separate jobs and may be initiated in any order. Once the control region is activated, messages may be input from terminals for processing by satellite regions that are already activated or currently not active (either not yet initiated or initiated but terminated via control terminal command or abend). When a satellite region is inactive, messages may be held in queues to await activation of this satellite region, or rejected, or routed to an alternate satellite region, based upon the subsystem's specification in the Region Descriptor Table. In this case, a message is returned to the terminal operator indicating disposition of the input message.

A series of special terminal commands (optionally restricted to the control terminal) allow operator control of the multiregion environment. Commands are available to effect the following operations:

- Terminate all or selected satellite regions via normal closedown (similar to the NRCD command) or immediate closedown (similar to the IMCD command). (The standard Intercomm closedown commands, NRCD and IMCD, are used to terminate both the control and all satellite regions, or only the control region.)

- Flush messages from interregion queues for all satellite regions, selected regions, selected subsystems within a region, or all subsystems.

- Stop or start message traffic to all satellite regions, selected regions, selected subsystems within a region, or all subsystems. Messages entered from terminals to a stopped region or subsystem are rejected; an appropriate message is sent to the terminal (see Section 2.1).

- Display the status of regions or subsystems, including a count of messages queued.

2.3.1 Message Restart

The normal Intercomm logging functions for messages within a region take place during system execution. Each region maintains its own separate log data set. Alternatively, a single log in the control region may be used by some or all satellite regions (indicated by coding MRlLOG parameter for the SPALIST for the region).

The control region log, if used for several regions, contains entries with duplicate message numbers. Using a single log eliminates the possibility for restart of either the control region or individual satellite regions, and should not be used when any data base or file updates of a critical nature are performed.

The control region log may optionally include entries for interregion message traffic, as well as for messages processed within the control region. The Region Descriptor Table specifies on a region and subsystem basis whether or not interregion log entries are made (LOG parameter). If log entries are made for interregion message traffic, restart of the control region restarts messages on the interregion and subsystem hold queues as well as standard (normal) restart of control region messages. All messages with log code A, without a corresponding log code of B, are restarted, if RESTART=YES is coded for the corresponding SUBSYS macro. (Messages on the interregion and/or subsystem hold queues are requeued for the destination, rather than physically preserving these queues.)

Control region log codes for multiregion entries are listed below:

Log Code	Meaning	Log by	CSECT
HEX C0	Region descriptor log entry logged at startup of each satellite region (from SR)	MRINTER	MRINTER
A (HEX C1)	Message successfully placed on satellite region Q by control region	MRQMNGR	MRQMON
B (HEX C2)	Message successfully passed to satellite region for processing	MRQMNGR	MRQMOFF
C (HEX C3)	Message not passed because satellite region/subsystem stopped or is inactive (Message on Hold Q if defined)	MRQMNGR	MRQMOFF

The RDT allows logging and restart specifications on a subsystem basis in a manner similar to its usual specification in SCTs. Synchronous logging may be specified. Restart may be specified as required or bypassed. If any satellite region terminates abnormally, it may be restarted via the standard Intercomm message restart facility, if it maintains its own message log.

2.3.2 File Recovery

If file recovery is to be performed (using Intercomm File Recovery), no other regions should access the same OS/VS files while message restart/file recovery is performed. For file recovery purposes, it is preferable that all subsystems that access a particular file are assigned to the same satellite region, whether or not inquiries or updates to the file are made. Then, should an abend occur in the satellite region, only that particular satellite region need be restarted.

If this is not feasible, all subsystems which update files should be assigned to the same satellite region; then, if an abend occurs in the update satellite region, only that satellite region need be restarted. However, it is safest to temporarily shut down inquiry satellite regions or use the FILE control command, since during file recovery, inquiry subsystems resident in the other satellite regions would risk less chance of retrieving invalid data. Alternatively, file-sharing inquiry subsystems in other satellite regions can be stopped by a multiregion control command during the time of the update region restart.

The worst situation occurs when two satellite regions update the same files. The satellite regions cannot update the same files while attempting to maintain file integrity at restart because both satellite regions require file recovery and restart if either region abends, and updates may not be processed in the same sequence as they originally occurred. Also, exclusive control is not possible when a file is updated from more than one region.

2.3.3 Data Base Processing and Recovery

When application subsystems communicate with a DBMS for access of information, similar considerations apply for region-oriented configuration of subsystems in the multiregion environment. If only inquiry-type access is performed in the entire system, user subsystems may be assigned to any satellite region or to the control region.

If any update-type access is performed in the system, all subsystems accessing the data base must be assigned to the same region. Then, in the event of abnormal termination of either the subsystem region or the data base manager region, only those two regions need be restarted. If separate satellite regions are required for separate application subsystems, and those satellite region subsystems access different data bases under the control of the same DBMS, multiple DBMS regions might be used as well. However, Intercomm's ability to support multiple DBMS regions depends on which DBMS is in use.

Some installations may need to operate with more than one DBMS (that is, while in the process of conversion from one data base management system to another), each operating independently in separate regions. In this case different satellite regions may access a different DBMS. The same considerations for assignment of subsystems to satellite regions apply, as discussed.

2.4 SUBSYSTEM REGION ASSIGNMENT

Based on the previous discussion of restart/recovery, subsystems are assigned to the various regions in the multiregion environment consistent with the following guidelines. (Figure 6 illustrates a typical grouping of subsystems by application area.)

2.4.1 Control Region Subsystems

Output Utility, Change/Display Utility, Multiregion Control Subsystem, Closedown Subsystem, Intercomm-supplied subsystems (File Handler Statistics subsystem, General Purpose Subsystem, 3270 Copy Subsystem, Checkpoint Subsystem and so forth) should be assigned to the control region. Any user subsystem with high-volume message traffic (such as a generalized editing and/or message routing subsystem) may also be assigned to the control region. Any subsystem that may receive messages from many other subsystems should be resident in the control region, to eliminate the need for either a copy of the subsystem in each user region or the transfer of messages from the satellite region to the control region and back to another satellite region.

2.4.2 Satellite Region Subsystems

Each satellite region must contain the Multiregion Output Subsystem, the Intercomm Closedown Subsystem, the Intercomm Checkpoint Subsystem (if file or data base updates are performed), and groups of user subsystems related to the same application. All subsystems

accessing the same OS/VS data sets should be grouped in the same satellite region if any subsystems update those data sets. All subsystems accessing a DBMS should be grouped in the same satellite region if data base updates are performed. Optionally, the Intercomm Output Utility may also reside in any satellite region. Use of this option is determined by whether or not a requirement exists for any region-specific Output Format Tables. (See the Utilities Users Guide.)

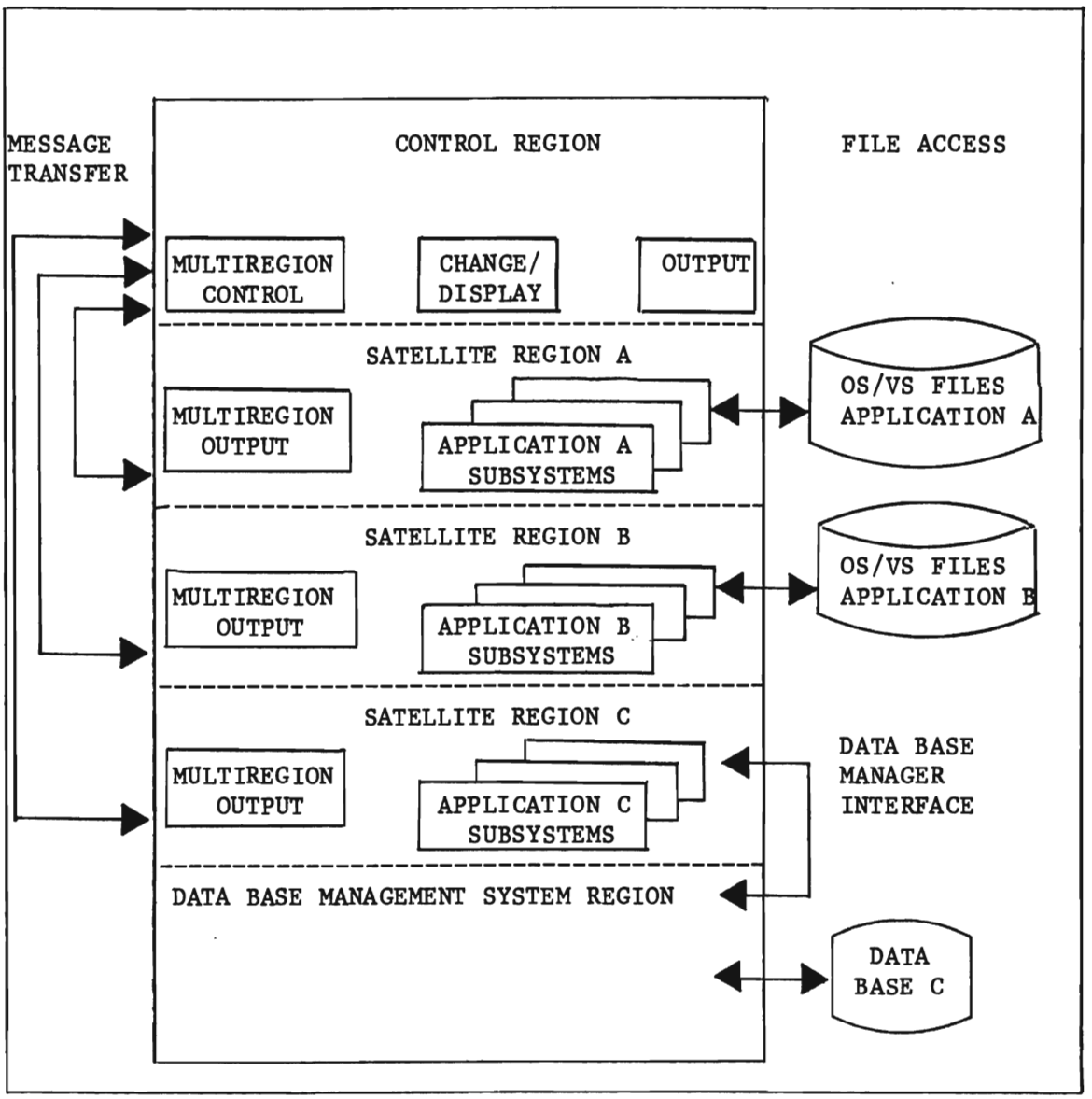

Figure 6. Satellite Region Subsystem Grouping by Application Area

2.5 SYSTEM COMMANDS

Intercomm offers a variety of system commands (processed by Intercomm subsystems) that control system operation on-line. System commands may be restricted to the control region, used in any one region, or required in more than one region. A subsystem that is required in multiple regions must have special definition in the Intercomm tables:

1. The Front End Verb Table needs multiple entries with different verbs specifying the unique subsystem code for each region.

2. Each region's Subsystem Control Table must have an entry for the subsystem. A different subsystem code must be used in each region.

3. The subsystem must also be described for each region in the Region Descriptor Table.

If Region Associated Processing (RAP) is used, multiple entries in the Verb Table and unique subsystem codes in the RDT are not necessary. (Refer to Chapter 8 for more detailed information on RAP.)

Certain command subsystems must be in the control region; other subsystems are verb-dependent (that is, test the message text for one of several specific verbs) and, thus, may be used only in one region. Figure 7 summarizes system commands and allowable region assignments. The Intercomm closedown commands are associated with the control region which subsequently notifies the satellite regions. Refer to System Control Commands for a detailed description of each command.

2.6 BATCH PROGRAM INTERFACE

A batch program may communicate with Intercomm subsystems operating in the multiregion environment by calling a supplied interface module to forward a message to Intercomm. This technique can be used to accomplish virtually any communication required from the batch application to an Intercomm application, such as:

- Forwarding a message to the control terminal to indicate that subsystems or files previously stopped by the operator may be started because a batch program's processing is complete.

- Forwarding a message to a terminal or subsystem to indicate that processing of a dynamic data queue created under Intercomm has completed, or that an output DDQ is complete and ready for processing by an Intercomm subsystem.

Command	Function	Region*
SWCH	Message Switching Subsystem	CR Only
SNBK	Echo Message Subsystem	CR or SR
NRCD/IMCD	System Closedown Subsystem	CR Only
TPUP/TDWN	Terminal Up/Down**	CR Only
STLG/SPLG	Start/Stop a line group**	CR Only
STLN/SPLN	Start/Stop line**	CR Only
SPPL/STPL	Stop/Start polling**	CR Only
LOCK/UNLK	Lock/Unlock verb**	CR Only
STAT	Display Front End Statistics**	CR Only
FLSH/RLSE	Flush/Release message from terminal queue**	CR Only
QHLD/QRLS	Hold/Release dedicated terminal queues**	CR Only
COPY	3270 Copy Subsystem	CR Only
MNCL DELY/BEGN	Fine Tuner Subsystem: Change MNCL, Stop/Start subsystem processing	CR or SR (One Region Only)
FHST	File Handler Statistics Subsystem	CR or SR
PGFX	Fix/Unfix VS Pages Subsystem	CR or SR
LOAD	Dynamic Load Control Subsystem	CR or SR
ASGN/DSGN	Activate/Deactivate Sign-on Security	CR Only
SECN/SECF	Activate/Deactivate control terminal security	CR Only
AVRB/DVRB SWON/SWOF	Activate/Deactivate system or terminal transaction security	CR Only
SNAP/ABND STRT/STOP FILE, TALY LTRC	General Purpose Subsystem: Issue Snap or abend, Start/Stop system or user function, OPEN or CLOSE file, System Statistics, Front End line trace	CR Only

**Front End verbs--not processed by a subsystem.
* CR = control region; SR = satellite region

Figure 7. System Commands and Assignments

2.7 MULTIREGION COMPONENTS

Apart from the standard Intercomm system components, such as the File Handler, Dispatcher, Resource Manager, Subsystem Controller, Logging/Restart/Recovery routines, a number of new components are required in a multiregion environment. The various modules are summarized in Figure 8, with a brief functional description. Detailed descriptions are provided in the following chapters.

Component	Residency*
MRMCT Multiregion Communications Table: User coded. Contains communications channels for interregion data switching.	LPA OR RAM FLPA-MVS
PMIRDTnn Region Descriptor Table: User coded. Defines satellite regions; their subsystems, queues, abend procedures, etc. (nn identifies table to load at startup).	CR Only
MRINPUT Multiregion Input Processor: Accepts, logs, and queues messages from satellite, control, and batch regions.	CR and each SR
MRSTAE Multiregion STAE routine: Called by STAEEXIT routine when satellite or control region abends.	CR and each SR
MRINTER Multiregion Initiation/Termination Processor: Handles startup, closedown, and abend processing of control and satellite regions.	CR and each SR
* CR = control region; SR = satellite region	

Figure 8. Multiregion Components (Page 1 of 2)

Component	Residency*
MROTPUT The Multiregion Output Subsystem: resident subsystem which sends messages from satellite regions to control region.	All SRs
MRCONSS Multiregion Control Subsystem: controls the multiregion environment from a terminal. May be defined as a resident or loaded subsystem.	CR Only
MRBATCH The Multiregion Batch Output Processor: accepts and sends messages from batch application programs to control region.	Batch Only
MRLOGIN and MRLOGOT The Multiregion Logging Routines: log accepted messages or send messages for logging. Optional component, required only if Single Log feature is to be used.	CR (MRLOGIN) and SR (MRLOGOT)
MRQMNGR Multiregion Queue Manager. Queues, dequeues, logs and transfers messages destined for satellite regions.	CR Only
KEYFLIP System component used in conjunction with IGCICOM.	CR and each SR
IGCICOM Type 1 SVC routine: Must be assembled and linkedited into user's OS/VS Nucleus. (Assigned SVC number is identified by the global &MRSVC in SETGLOBE.)	OS/VS Nucleus
MRPURGE Closedown purge processing.	CR and each SR
MRCSAMOD CSA processing-interregion message transfer-MVS only.	CR and each SR
MRMOD Region Associated Processing (RAP) command subsystem for LOKR/ULKR commands.	CR only
* CR = control region; SR = satellite region	

Figure 8. Multiregion Components (Page 2 of 2)

Chapter 3

CODING MULTIREGION TABLES

Multiregion tables are coded using Intercomm macros developed for this facility. Residency of these tables depends on their function, as follows:

- The Multiregion Communications Table (MCT) resides in the Link Pack Area (OS/MVT or VS/2), Resident Access Method area (OS/MFT or VS/1), or Fixed Link Pack Area (MVS). The MCT defines all potential user region identifications and provides for interregion communication ECB channels.

- The Region Descriptor Table (RDT) resides in the control region only. However, it is loaded based on a WTOR reply by the console operator at system startup. Thus, the user may define many operational region configurations and load the appropriate table on demand at startup time.

- The System Parameter Area (SPA) resides in the control region and in each satellite region. It defines the System Parameter List and the Subsystem Control Table (SCT) entries for that region.

Batch regions sending messages to the Intercomm control regions contain no multiregion tables. All table-oriented macros associated with Multiregion Support are described in this chapter.

3.1 MULTIREGION COMMUNICATIONS TABLE (MCT)

This table, consisting of region identifiers and ECB channels, is generated by coding one REGCOM macro for every possible satellite region that may be initated and one REGCOM macro if any batch regions may send messages to Intercomm.

The CSECT name generated by the assembly is MRMCT. The load module member name is MRMCT. See Appendix A for the coding of the REGCOM macro.

The JCL required to create, assemble and linkedit the table is illustrated below. The load module must then be added to the user's Link Pack Area, Resident Access Method area, or Fixed Link Pack Area, as applicable for the operating system in use.

```
//GENMCT  EXEC LIBELINK,Q=xxx,NAME=MRMCT,LMOD=MRMCT,
//        PARM.LKED='LIST,LET,DC,RENT'
//LIB.SYSIN     DD *
./ADD     NAME=MRMCT,LIST=ALL
***   MULTIREGION COMMUNICATIONS TABLE
***   CSECT IS INTERNALLY GENERATED ***
          REGCOM   RGNID=xxxxxxxx  SATELLITE REGION X
          REGCOM   RGNID=yyyyyyyy  SATELLITE REGION Y
           .
           .    other satellite regions
           .
          REGCOM   RGNID=BATCH     BATCH REGIONS
          END
```

MRMCT may not be contained in the STEPLIB or JOBLIB of any region, so that one copy of MRMCT is shared by all regions.

The table entry generated by each REGCOM macro contains the ECB channels used to effect interregion communication. An ECB channel is defined as a pair of ECBs, the first used to request a function, the second to acknowledge the processing of the request. Every MCT entry has the following channels:

- Status Channel

  Control region Status ECB is posted by the control region when it starts, terminates or abends. Satellite region Status ECB is posted by the satellite region when it starts, terminates or abends.

- Input Channel

  Input ECB is posted by the control region with the address of a message to be sent to a satellite region. Input acknowledge ECB is posted by the satellite region after queuing the message received via the Input ECB.

- Output Channel

  Output ECB is posted by the satellite region with the address of a message to be sent to the control region. Output Acknowledge ECB is posted by the control region after queuing the message received via the Output ECB.

- Asynchronous Logging Channel

  Asynchronous Logging ECB is posted by the satellite region with the address of a message to be logged asynchronously (see SYCTTBL macro LSYNCH parameter). Asynchronous Logging Acknowledge ECB is posted by control region to indicate completion of asynchronous logging request.

- Synchronous Logging Channel

  Synchronous Logging ECB is posted by the satellite region with address of message to be logged synchronously (see SYCTTBL macro LSYNCH parameter). Synchronous Logging Acknowledge ECB is posted by control region to indicate completion of synchronous logging request.

The logging channels are used only if Single Region Logging is specified for a satellite region.

3.2 REGION DESCRIPTOR TABLE (RDT)

The RDT, a required table that must reside only in the control region, is generated by coding REGION and SUBSYS macros, plus one GENRDT macro. There must be one REGION macro for each satellite region and one SUBSYS macro for each subsystem in each satellite region that receives messages from terminals or other satellite regions (via the control region). The control region and batch regions are not defined in the RDT.

All satellite region subsystems to receive messages from terminals or from subsystems in another region must be defined by SUBSYS macros in the RDT. (If a subsystem is defined in the RDT but is not present in the satellite region SCT, a Snap 127 is issued in the satellite region, see Figure 3.) Duplicate subsystem codes are not allowed unless RAP is implemented (see Chapter 8). Subsystems that only receive messages from subsystems in the same region need not be defined in the RDT. If a subsystem is present in the region, but the region is not the primary region for that subsystem (that is, this region is the alternate region for the subsystem), a SUBSYS RDT entry must not be defined for this subsystem within this region. Also, the Intercomm closedown subsystem, although residing in each satellite region, may not be defined in the RDT. Neither may the checkpoint nor the MROTPUT susystems be defined in the RDT.

The RDT is loaded dynamically by the control region at system initialization. The member name of the generated table must be PMIRDTnn (where nn is any two characters). The nn in the member name identifies the table that is used for a particular execution of the Intercomm system and is the reply to a WTOR issued at system startup. The default reply to the WTOR is '00'. See Appendix A for coding of the REGION, SUBSYS, and GENRDT macros.

The JCL required to create, assemble and linkedit the Region Descriptor Table is illustrated in Figure 9. The linkedit JCL for the table must specify ENTRY PMIRDT. The symbolic form of the table then resides on the library PMI.SYMLIB; the load module is on PMI.MODLIB. The load module library must be defined by STEPLIB or JOBLIB at execution time. The load module may not be included in the Intercomm linkedit.

```
//GENRDT     EXEC LIBELINK,Q=LIB,NAME=PMIRDTnn,LMOD=PMIRDTnn
//LIB.SYSIN  DD *
./ ADD       NAME=PMIRDTnn,LIST=ALL
****         MULTIREGION DESCRIPTOR TABLE ****
****         IDENTIFIER OF REGION CONFIGURATION IS nn ****
****         CSECT IS INTERNALLY GENERATED ****
             REGION  xxxxxxxx,...
             SUBSYS  x,1,...
             SUBSYS  x,2,...
               :
               :
             REGION  yyyyyyyy,...
             SUBSYS  y,1,...
             SUBSYS  y,2,...
               :
               :
             GENRDT  (No parameters.  Must be the last macro.)
             END
//LKED.SYSIN DD *
             ENTRY PMIRDT              (Required control card)
             NAME PMIRDTnn(R)
```

Figure 9. Coding an RDT

3.3 REGION DESCRIPTOR TABLE WITH RAP

If Region Associated Processing is used, the MRPASWRD macro is coded to associate a particular region with particular passwords. These passwords are also associated with particular terminals via Front End Table BTERM/LUNIT/LCOMP macro coding. For detailed specifications on MRPASWRD macros and the additional parameter on the terminal macros, see Chapter 8.

3.4 SYSTEM PARAMETER AREA

The control region and each satellite region must contain a System Parameter Area, and Subsystem Control Table (SCT) entries for each subsystem in the region. The SPALIST macro defines the System Parameter Area; the SYCTTBL macro defines the SCT entries. Refer to Basic System Macros for complete specifications.

Multiregion control parameters for the SPALIST macro are as follows:

[symbol]	SPALIST	...(other parameters)... [,MRAUTO={WAIT}] [{ENDN}] [{DUMP}] [{NO }] [,MRCNTL={YES}] [{NO }] [,MRCSALN={nnnnn}] [{1024 }] [,MRID=xxxxxxxx] [,MR1LOG={YES}] [{NO }]

MRAUTO

specifies, for satellite regions, the action to be automatically taken by this satellite region if it detects that the control region is down. If WAIT is specified, the satellite region goes into a wait state until the control region is restarted. If DUMP is specified, the satellite region terminates with a 556 abend and a dump. If ENDN is specified, the satellite region terminates with a 555 abend and no dump. If none of the above options is specified, message RC004R (which requests one of these options) is sent to the CPU console operator. NO, the default, specifies that the action to be taken is to be specified via the operator response to RC004R.

MRCNTL

specifies whether or not this is the System Parameter Area for the control region. If for the control region, code YES. The default is NO, indicating it is for a satellite region.

MRCSALN

specifies, for satellite regions under MVS, the number of bytes of CSA to acquire at startup time and hold until region termination. The value specified is rounded down to the next lower multiple of 8 if it is not already a multiple of 8. The core acquired is used when sending messages from this region to the control region. Therefore, it should be as large as the average message sent to the control region. (If a message is larger than the CSA specified, more CSA is temporarily acquired to handle it.) Code as a decimal value, 48 to 32760. The default is 1024.

MRID

specifies, for satellite regions, the region identifier, which must be unique and must also be coded in the MRMCT table via the RGNID parameter of the REGCOM macro, and in the RDT table via the REGION macro. Must be coded as 1 to 8 alphameric characters. MRID=CONTROL may not be specified.

MRILOG

specifies, for satellite regions, whether or not logging for this region is to occur in the control region. The default is NO, indicating a separate log. Single region logging involves a lot of overhead and precludes the use of message restart, checkpointing, or file or DBMS recovery for the satellite region. It is not recommended unless very little logging of satellite region activity is specified. (LOG=NO coded for most SYCTTBLs).

The member SETGLOBE, which controls assembly of the SPALIST macro, must contain settings as follows:

```
&MULTREG    SETB        1          Multiregion Facility in use
&MRSVC      SETC        'xxx'      xxx is the number for the
                                   Multiregion SVC
```

3.5 SUBSYSTEM CONTROL TABLE (SCT)

Subsystem Control Table entries (via the SYCTTBL macro) are required in each region as described below. There are no SYCTTBL macro parameters specifically associated with Multiregion Support.

3.5.1 Control Region SCT Entries

1. An SCT entry for the Output Utility (PMIOUTPT) must be defined (subsystem codes U, V, N).

2. An SCT entry must be defined for the closedown subsystem (subsystem code J) which processes the NRCD and IMCD commands.

3. An SCT entry for the Multiregion Control Subsystem (MRCONSS) may be defined only in the control region.

4. An SCT entry must be defined for each subsystem in the control region which processes system messages and commands received from terminals (such as GPSS, etc.).

5. An SCT entry must be defined for each subsystem in the control region which processes messages received from a satellite region (such as the Output Utility).

6. SCT entries may be defined for user application subsystems.

3.5.2 Satellite Region SCT Entries

1. An SCT entry for the Multiregion Output Subsystem (MROTPUT) must be defined in each satellite region (Subsystem code Z). This subsystem is not defined in the RDT.

2. An SCT entry must be defined for each subsystem receiving messages from the control region (input from terminals or other satellite regions) or other subsystems within the same satellite region.

3. An SCT entry must be defined for the closedown subsystem (Subsystem code J). This subsystem is not defined in the RDT.

4. An SCT entry should be defined for each subsystem defined as an alternate to a subsystem in a different satellite region. (See SUBSYS macro, ALT parameter).

5. Definition of the Output Utility in any satellite region is an option which allows the use of region-associated Output Format Tables. (See the Utilities Users Guide). This subsystem is not defined in the RDT.

Subsystem codes defined by SCT entries in each region must be unique in the Intercomm system except in the following cases:

- If RAP is in use.
- If the subsystem receives messages only from other subsystems within the same region (no SUBSYS macro is coded in the RDT).
- If the subsystem is to act as the alternate subsystem when the primary subsystem (region) is inactive. (In this case, no SUBSYS macro is coded for this subsystem in the RDT entry defining this region.)
- If the subsystem is the closedown, checkpoint, or MROTPUT subsystem.

The JCL required to define, assemble and linkedit the System Parameter Area and SCT for the control region and a satellite region is illustrated in Figures 10 and 11. The Intercomm SPA Extension must be separately assembled and linkedited for each region; the required CSECT names are SPA and SPAEXT, respectively. The SPA and SCT may be separately assembled, if necessary, as described in the Operating Reference Manual.

```
//CRSPA         EXEC  LIBELINK,Q=LIB,NAME=CRSPA,LMOD=CRSPA
//LIB.SYSIN     DD    *
./ ADD NAME=CRSPA
*** CONTROL REGION SPA AND SCTS
SPA  CSECT
     SPALIST (other parameters)...,MRCNTL=YES
     COPY SCTLISTC
* OUTPUT UTILITY SCTS
     SYCTTBL SUBC=U,SBSP=PMIOUTPT,...
     SYCTTBL SUBC=N,SBSP=PMIOUTPT,...
     SYCTTBL SUBC=V,SBSP=PMIOUTPT,...
* MULTIREGION CONTROL SUBSYSTEM SCT
     SYCTTBL SUBC=K,SBSP=MRCONSS,...
* OTHER SUBSYSTEMS AS REQUIRED NEED SCTS
     SYCTTBL SUBC=H,SBSP=CHANGE,... (Change/Display)
     SYCTTBL ...
     .          user subsystems, system control command
     .          subsystems, Closedown subsystem, etc.
     .
     SYCTTBL ...
* GENERATE SCTINDEX
     GENINDEX
     PCENSCT
     END
//CRSPAEXT      EXEC LIBELINK,Q=LIB,NAME=CRSPAEXT,LMOD=CRSPAEXT
//LIB.SYSIN     DD *
./ ADD NAME=CRSPAEXT
*** CONTROL REGION SPA EXTENSION
SPAEXT CSECT
     SPALIST EXTONLY=YES,...(identical parameters as in SPALIST above)
     END
```

Figure 10. Coding the Control Region SPA, SCT and SPAEXT

```
//SRSPA         EXEC LIBELINK,Q=LIB,NAME=SRSPA,LMOD=SRSPA
//LIB.SYSIN     DD *
./ ADD NAME=SRSPA
*** TYPICAL SATELLITE REGION SPA AND SCT
SPA     CSECT
        SPALIST (other parameters)...,MRID=SATREGNA
        COPY SCTLISTC
* MULTIREGION OUTPUT SUBSYSTEM
        SYCTTBL SUBC=Z,SBSP=MROTPUT,MNCL=1,
                LANG=RBAL,PRTY=0,TCTV=0,
                OVLY=0, (must be resident)
                .
                .    other operands
                .
*OUTPUT UTILITY SUBSYSTEM
*DEFINED AS AN OPTION
        SYCTTBL SUBC=U,SBSP=PMIOUTPT,...
        SYCTTBL SUBC=V,SBSP=PMIOUTPT,...
        SYCTTBL SUBC=N,SBSP=PMIOUTPT,...
        SYCTTBL ...
        .       user subsystems,
        .       Closedown subsystem, etc.
        .
        SYCTTBL ...
* GENERATE SCT INDEX
        GENINDEX
        PCENSCT
        END
//SRSPAEXT      EXEC LIBELINK,Q=LIB,NAME=SRSPAEXT,LMOD=SRSPAEXT
//LIB.SYSIN     DD *
./ ADD NAME=SRSPAEXT
*** TYPICAL SATELLITE REGION SPA EXTENSION
SPAEXT CSECT
    SPALIST EXTONLY=YES,...(identical parameters as in SPALIST above)
    END
```

Figure 11. Coding the Satellite Region SPA, SCT and SPAEXT

Chapter 4

MULTIREGION PROCESSING ROUTINES

The following provides an overview of the logic of the major components of the Multiregion Support Facility. It is assumed that the reader is familiar with the Intercomm DISPATCH macro and associated control of multithreading via the System Dispatcher.

4.1 MULTIREGION INITIATION/TERMINATION PROCESSOR (MRINTER)

The module MRINTER, whose function is to handle region startup, closedown, and abend processing, must be present in all Intercomm regions, both control and satellite. This module comprises three control sections: MRSTART, MRCLOSE, MRINTER. The first two may reside in the startup and closedown overlays; the third must be resident. The function of each control section is discussed below.

4.1.1 MRSTART

Called by the Intercomm startup module STARTUP3, this control section stores both the address of the current TCB (ASCB for MVS) and the address of the region's entry in the MCT within a dynamically obtained work area. MRSTART issues an exclusive ENQ (with rname of the region ID from the MCT) to permit monitoring by the monitor subtask, MRASYNCH, which is attached as a subtask to monitor a region's activity.

MRINTER is dispatched by MRSTART and is passed the address of the dynamic work area. In the control region, MRSTART dispatches MRINTER for every region entry in the Region Descriptor Table and stores the RDT entry address in the work area. MRINTER is dispatched only once in a satellite region. If batch regions are defined in the RDT, MRSTART in the control region dispatches MRINPUT to handle batch region input.

4.1.2 MRCLOSE

This control section, called by the Intercomm closedown module, posts the region's status ECB with the closed status code, then detaches the monitor subtask and DEQs from its own region-ID.

4.1.3 MRINTER

This control section is dispatched by MRSTART and passed the address of the dynamic work area.

In the control region, MRINTER is dispatched for every satellite region defined in the RDT. In the control region, MRINTER initializes itself to handle a particular satellite region, then posts its status ECB for the satellite region and waits for the satellite region, in turn, to post its status ECB. After the ECBs are posted, the satellite region is marked active in the RDT. A monitor subtask, MRASYNCH, is attached to monitor the satellite region. An MRINPUT thread is dispatched to receive input from the satellite region. If it is included in the linkedit, MRLOGIN is dispatched to handle single-region logging requests from the satellite region. MRINTER then waits for the status ECB of the satellite region to be posted with a termination code indicating whether the satellite region has closed or abended. When termination of the satellite region is posted, the control region notifies the control terminal operator, deactivates the region in the RDT, and waits for the satellite region to post its status ECB indicating the satellite region has been restarted.

MRINTER is dispatched only once in a satellite region. In a satellite region MRINTER issues an in-line WAIT on the status ECB of the control region. When the control region becomes active, it is flagged as active in the System Parameter Area. A monitor subtask, MRASYNCH, is attached to monitor the control region. MRINPUT is dispatched to receive input from the control region. MRINTER then waits for the status ECB of the control region to be posted with a termination code indicating whether it has closed or abended. When termination of the control region is posted, the satellite region executes its SPALIST MRAUTO parameter request, or if NO, it issues a WTOR requesting the OS/VS Console Operator to reply WAIT, DUMP, or ENDN. A WAIT request causes the satellite region to wait for the control region restart by issuing an in-line WAIT on the control region's status ECB. DUMP and ENDN requests cause the satellite region to abend, the former command producing a dump.

4.1.4 Monitor Subtask (MRASYNCH)

MRASYNCH (an entry in MRINTER) monitors the activity of a region in such instances as a region terminating without notifying other regions (that is, without posting its status ECB). MRASYNCH enqueues upon the other region-ID (control or satellite, as appropriate), thus putting the subtask into the WAIT state. If the enqueue completes, this indicates the other region has terminated without posting its status ECB. MRASYNCH then immediately dequeues, posts the status ECB of the host region on behalf of the region that terminated, and exits.

4.2 MULTIREGION INPUT PROCESSOR (MRINPUT)

The module MRINPUT must be present in all Intercomm regions, both control and satellite. The module is dispatched by the Multiregion Initiation/Termination Processor (MRINTER) each time a region is activated so that the activated region can receive messages. MRINPUT moves a message from the issuing region to the receiving region's own dynamic storage area. After the module accepts messages in the receiving region it passes them to the Intercomm Message Collection Routine, (or FESEND, if appropriate, in the control region).

In the control region, multiple threads of the MRINPUT module exist: one for every satellite region that is active and one thread for all batch region input. In a satellite region, only a single MRINPUT thread is active, since only the control region can input messages to the satellite region. After initialization, MRINPUT dispatches itself to wait for an input ECB to be posted, after which MRINPUT saves and then clears the contents of the ECB. If the saved post code (bits 2-7 in the high-order byte of the ECB) is nonzero, MRINPUT terminates, thus deactivating the thread. If the saved post code is zero, it moves the message whose address is in the low-order 3 bytes of the ECB to its own dynamic storage. MRINPUT calls Message Collection to log and queue the message, then posts the input acknowledge ECB for the issuing region to indicate that the message was received. Then MRINPUT dispatches itself again to wait for an input ECB posting.

4.3 MULTIREGION QUEUE MANAGER (MRQMNGR)

This module, MRQMNGR, resides in the control region of the multiregion environment and processes queuing, dequeuing, and transfer of messages destined for satellite regions. MRQMNGR has four entry points, MRQMON, MRQMOFF, MRQMGET and MRQMSUB, discussed below.

4.3.1 MRQMON

The entry point MRQMON is called by Message Collection whenever it determines that the subsystem code in the message is not defined in any SYCTTBL resident in the control region. MRQMON logs the message as it is received and attempts to queue it. If unable to queue the message, MRQMON returns to Message Collection with an error return code. Message Collection then determines the appropriate action. If the message is successfully queued, and MRQMOFF is still inactive, MRQMON dispatches MRQMOFF and returns to Message Collection.

4.3.2 MRQMOFF

The entry point MRQMOFF is dispatched by MRQMON or by MRINTER whenever a satellite region is started. MRQMOFF is passed the address of a region entry in the RDT and first checks the region for any subsystem hold queues. If there are any, MRQMOFF dispatches MRQMSUB to dequeue and transfer all messages on the subsystem hold queues to the satellite region. Until MRQMSUB is finished, MRQMOFF waits on an ECB posted by MRQMSUB. MRQMOFF then dequeues a message from the region queue and transfers it to the satellite region by posting the input ECB of the satellite region with the message address. When operating with MVS, the message is first moved to an area in CSA. MRQMOFF repeats this procedure until the region queue is exhausted, and then terminates.

4.3.3 MRQMGET

The entry point MRQMGET is dispatched by MRQMOFF whenever the latter detects both an empty core queue and messages present on a disk overflow queue. MRQMGET primes all core queues and attempts to keep them filled at all times. MRQMGET reads a message from the disk component of the region queue and stores it in the core component. Multiple MRQMGET threads may be dispatched to refill empty core queue elements. While MRQMGET is filling a core queue element, the element is marked as having I/O pending. If MRQMOFF encounters the I/O pending indicator in a core queue element, it waits for MRQMGET to post an ECB indicating that the I/O is complete.

4.3.4 MRQMSUB

The entry point MRQMSUB is dispatched by MRQMOFF whenever subsystem hold queues exist. MRQMOFF creates hold queues after dequeuing a message from the region queue and detecting that the region is inactive. Subsystem hold queues are created only for subsystems with no active alternate regions and for which hold queues have been specified via the IFDOWN parameter. MRQMSUB dequeues messages from each subsystem hold queue until all are empty, passing the messages to the satellite region. When all subsystem hold queues are empty, MRQMSUB posts an ECB on which MRQMOFF is waiting and deactivates itself.

4.4 CONTROL REGION SINGLE LOG INPUT (MRLOGIN)

MRLOGIN is to be included only if the Single Log feature is being used, and must be resident in the control region. MRLOGIN contains two entry points, MRLOGASY and MRLOGSYN, which are dispatched by MRINTER when a satellite region starts up; the first entry point handles asynchronous logging requests from the satellite region, the second, the synchronous requests.

4.4.1 MRLOGASY

When posted with a message to log, MRLOGASY copies the message from the satellite region to the control region, dispatches an entry point in MRLOGIN (MRLOGDSP), posts an acknowledgement ECB to indicate receipt of the logging request, and dispatches itself to wait for the next logging request. When given control, MRLOGDSP calls LOGPUT to do an asynchronous write to the log and then frees the message and exits.

4.4.2 MRLOGSYN

When posted with a message to log, MRLOGSYN copies the message into the control region and calls LOGPUT to write the message to the log synchronously. When LOGPUT returns, MRLOGSYN frees the message and posts an ECB to indicate that the request was processed. It then dispatches itself to wait for the next synchronous logging request.

4.5 SATELLITE REGION SINGLE LOG OUTPUT (MRLOGOT)

The module MRLOGOT must reside in the satellite region or regions requesting the Single Log feature. In the satellite region, LOGPUT calls MRLOGOT after determining that log entries are to be written on the control region log. Under MVS, the message is moved to an area in the CSA. If the log entry is to be asynchronous, MRLOGOT posts an appropriate logging ECB with the address of the log entry and issues an OS/VS WAIT macro on an acknowledgement ECB. After posting, MRLOGOT returns to LOGPUT. At this point, the log entry has been received by the control region. However, the log entry may or may not have been written to the log.

If logging is to be synchronous, MRLOGOT enqueues on the synchronous logging channel and posts the channel with the address of the log entry. It then does a dispatch WAIT on an acknowledgement ECB. After the ECB is posted, MRLOGOT dequeues off the channel, frees the CSA area if on an MVS system, and returns to LOGPUT. At this point the log entry has been written.

If the acknowledgment ECB is posted with a nonzero post code, or if the control region is inactive, MRLOGOT waits upon an internal ECB (SEXMRECB) to be posted by MRINTER, indicating that the control region is up. It then retries the logging operation.

If MRLOGOT is included, PMISNAP1 (ICOMSNAP CSECT), containing the entry points CTIMER and STIMER, should also be made resident in the associated satellite region.

4.6 MULTIREGION OUTPUT SUBSYSTEM (MROTPUT)

MROTPUT is a standard Intercomm subsystem, and must be defined as a resident subsystem in all satellite regions. MROTPUT sends messages from satellite regions to the control region.

Under MVS, MROTPUT moves the message to the CSA and, after enqueuing on the output channel, posts the output ECB of the satellite region with the address of the message it is sending. MROTPUT then waits for the control region to post the output acknowledgement ECB, indicating receipt of the message. After posting, MROTPUT dequeues off the output channel, frees the CSA (if any), and returns to the Subsystem Controller.

For message transfer, subsystem logic need not be concerned with the region residency of any other subsystems. Subsystems create messages and identify the destination subsystem in the message header. Message Collection automatically reroutes messages to Multiregion Output if the destination subsystem for a message being queued is not defined in that satellite region Subsystem Control Table.

The SYCTTBL macro for MROTPUT must be coded as follows:

```
SYCTTBL SUBC=Z,LANG=RBAL,PRTY=0,SBSP=MROTPUT,TCTV=0,OVLY=0,...
```

Other parameters may be coded as necessary. MNCL=1 is advised.

For terminal output, the subsystem calls FESEND (FESENDC) with all VMI 57 or 67 messages. The receiving subsystem code (if any) will be set (changed) to that of MROTPUT, and the message will be queued for MROTPUT for transfer to the control region. In the control region, MRINPUT recognizes the special RSC code and calls FESEND to queue the message for the terminal (broadcast group) designated by MSGHTID in the message header. Therefore, the MROTPUT RSC/H is reserved for MRS use when executing Intercomm under MRS.

Chapter 5

MULTIREGION BATCH INTERFACE

A batch application program executing in the multiregion environment may send a message to an Intercomm subsystem or to a user application subsystem executing either in a satellite region or the control region. For example, a batch program might send a message to a user subsystem to indicate completion of batch program processing. Also, a batch program may direct a message to a terminal by forwarding a message to the Intercomm Output Utility.

5.1 MRBATCH

The module MRBATCH must reside in any batch region that sends messages to an Intercomm subsystem or terminal. The batch program must be linkedited with MRBATCH prior to execution. MRBATCH locates the common batch region ECB channels in the MCT and does a system-wide enqueue on the output channel to prevent different batch regions from simultaneously using the channel. If under MVS, MRBATCH first moves the message to an acquired area in CSA. MRBATCH then posts the output channel with the address of the message, and waits for Intercomm to acknowledge its receipt. MRBATCH then dequeues off the channel, frees the CSA area (if under MVS), and returns to its caller with a return code indicating the status of the operation.

5.2 CODING CONVENTIONS

When a batch application program sends a message to Intercomm, it issues a call, for Assembler Language, COBOL and PL/1, as follows:

Assembler Language:

```
[label]   CALL        MRBATCH,(msg,return-code),VL
```

COBOL:

```
          CALL        'MRBATCH' USING msg, return-code
```

PL/1:

```
          CALL        MRBATCH  (msg, return-code);
```

where

- msg is the label (or address) of the message to send. The message must be half-word, fullword, or double-word boundary aligned.

- return-code is the label (or address) of a one-byte field for the return code in character format.

5.3 MESSAGE HEADER

The message passed to Intercomm must have a valid Intercomm message header (see Figure 12).

Certain fields in the message header specify message destination by receiving subsystem code. For example, to send a message to a terminal, via the Output Utility, use:

```
MSGHRSCH=X'00'
MSGHRSC=C'U'
```

MSGHTID identifies the destination terminal (or Broadcast group).

The message text format is identified by MSGHVMI (preformatted or Variable Format text). If a Fixed Format text is used, the message must be routed to the Change/Display Utility (MSGHRSCH=X'00', MSGHRSC=C'H', MSGHVMI=X'72'). VMI 57/67 messages may bypass the Output Utility for terminal queuing by coding the MROTPUT RSC of C'Z'.

The following fields must be initialized:

MSGHLEN--message length
MSGHRSCH—high-order byte of receiving subsystem code
MSGHRSC—low-order byte of receiving subsystem code
MSGHTID--a valid terminal ID
MSGHQPR--indicates a full message (C'2')
MSGHVMI—predefined codes for the receiving subsystem (or FESEND)

For detailed specifications, see the Assembler Language Programmers Guide, the COBOL Programmers Guide, or the PL/1 Programmers Guide.

5.4 RETURN CODES

The return codes from MRBATCH to the user program are as follows:

C'0'= normal completion
C'1'= Intercomm not active
C'2'= invalid message; Intercomm Queue Management not able to pass this message to an Intercomm subsystem (Message Collection return code nonzero).

Field Name	No. of Bytes	Description
MSGHLEN	2	Length of message including header (binary halfword number)
MSGHQPR	1	Full message identification
MSGHRSCH	1	Receiving subsystem code high-order byte
MSGHRSC	1	Receiving subsystem code low-order byte
MSGHSSC	1	(Sending subsystem code low-order byte)
MSGHMMN	3	(Monitor message number (binary) assigned by Message Collection)
MSGHDAT	6	(Julian date (YY.DDD) initialized by Intercomm)
MSGHTIM	8	(Time stamp (HHMMSSTH) initialized by Intercomm)
MSGHTID	5	Terminal identification (destination terminal or broadcast group)
MSGHCON	2	(Reserved)
MSGHPID	5	(Reserved)
MSGHSSCH	1	(Sending subsystem code high-order byte)
MSGHUSR	1	Available to user; C'P' indicates a priority queue message
MSGHBMN	2	(BTAM message number assigned by Front End)
MSGHLOG	1	(Log code assigned by Intercomm)
MSGHBLK	1	(Reserved)
MSGHVMI	1	Verb or message identifier interpreted by receiving subsystem as required

Figure 12. Fields in the Message Header for MRS Batch Interface

Chapter 6

MULTIREGION CONTROL SUBSYSTEM

The Multiregion Control Subsystem (MRCONSS) is assigned to the control region, but it may be a resident, overlay, or dynamically loaded subsystem. MRCONSS processes commands entered from a terminal (or optionally the control terminal only), to control the operation of the satellite regions. The subsystem also accepts commands in the form of a message routed from application subsystems. Any subsystem may route a message to MRCONSS; if control terminal security is specified for the command verb, the Destination Terminal Identification (MSGHTID) in the message header must be that of the defined control terminal.

The Multiregion Control Subsystem requires table definitions in the Verb Table (BTVRBTB) and Subsystem Control Table for the control region. Any verb may be used for the Multiregion Control Subsystem; the Intercomm standard is COMM. The commands are issued in subset form under this verb. If another verb is chosen, system routines must be modified (see Section 7.5.) The following is the coding for the Verb Table:

```
[label]   BTVERB     VERB=COMM,SSC=K[,SECUR=YES]
```

Several required operands for the SCT entry are defined as follows:

```
[label]   SYCTTBL    SUBC=K,SBSP=MRCONSS,LANG=NBAL,MNCL=1,
                     RESTART=NO,...other parameters as required
```

6.1 MULTIREGION CONTROL COMMANDS

The multiregion control commands allow certain actions to be applied to all, or selected, satellite regions, or all, or selected, satellite region subsystems. A maximum of 10 satellite region identifiers or subsystem codes may be entered with each command. Refer also to Chapter 8 for information on use of multiregion control commands with RAP. The commands are entered in the following formats:

Command Format	Function
vvvv$DOWN${ALL }@ {rrrrrrrr[$rrrrrrrr...]}	NRCD (Normal Closedown) all regions or selected regions
vvvv$QDOWN${ALL }@ {rrrrrrrr[$rrrrrrrr...]}	IMCD (Immediate Closedown) all regions or selected regions
vvvv$FLUSH${ALL }@ {rrrrrrrr[$rrrrrrrr...]}	Flush subsystem hold and interregion queues for all or selected regions
vvvv$FLUSH$SS${ALL }@ {h,c[$h,c...]}	Flush subsystem hold queues for all or selected subsystems
vvvv$STOP${ALL }@ {rrrrrrrr[$rrrrrrrr...]}	Stop input to all or selected regions
vvvv$STOP$SS${ALL }@ {h,c[$h,c...]}	Stop input to all or selected subsystems
vvvv$START${ALL }@ {rrrrrrrr[$rrrrrrrr...]}	Allow input to all or selected regions
vvvv$START$SS${ALL }@ {h,c[$h,c...]}	Allow input to all or selected subsystems
vvvv$STATUS${ALL }@ {rrrrrrrr[$rrrrrrrr...]}	Display Region Status
vvvv$STATUS$SS${ALL }@ {h,c[$h,c...]}	Display Subsystem Status

where:

- vvvv is the verb for the Multiregion Control Subsystem (that is, COMM).
- ALL indicates the command is to apply to all satellite regions, or all satellite region subsystems (if it is preceded by SS).
- rrrrrrrr is a satellite region identifier. Up to 10 regions may be specified, separated by system separator characters.
- SS indicates that the command is to apply to subsystems, rather than regions.

- h,c is a subsystem code, in which h is the high-order byte and c is the low-order byte, entered as a one-character alphameric code, its hexadecimal equivalent, or its decimal equivalent (applies only to satellite region subsystems).
- $ is the System Separator Character.
- @ is the End-of-Transmission Sequence.

6.2 COMMAND RESPONSES

Messages are returned to the terminal entering multiregion control commands as follows:

Response	Meaning
COMMAND SUCCESSFULLY PROCESSED.	Indicates message successfully processed by MRCONSS.
MESSAGE LENGTH INVALID. NO ACTION.	Indicates the input message was too long, with probably more than ten region or subsystem IDs entered.
INVALID COMMAND TYPE. NO ACTION.	Syntax error.
MISSING OR MISPLACED DELIMITER. NO ACTION.	Separator character not found in proper place.
INVALID REGION-ID. NO ACTION.	One or more Region-ids incorrect.
INVALID SUBSYSTEM CODE. NO ACTION.	One or more Subsystem codes incorrect, not found or not in password-associated region.
THE REGION DESCRIPTOR TABLE IS MISSING. NO ACTION.	RDT was not loaded at at startup.
*INVALID PASSWORD. NO ACTION.	Password not found in password table in RDT.
*PASSWORD TABLE MISSING. NO ACTION.	The RDT loaded at startup does not contain a password table (no MRPASWRD macros).
*A SUBSYSTEM IS NON-UNIQUE -- PASSWORD REQUIRED. NO ACTION.	Duplicate subsystem code found in RDT, but no password in command.

*These messages refer to RAP processing/command requirements. See Chapter 8.

Figures 13 and 14, respectively, show the types of messages returned in response to a STATUS Command requesting region status and to a STATUS command requesting subsystem status. The operator may use the Intercomm standard RLSE command to retrieve the next message in a series when 'MORE' is indicated on the resulting display. In the following figures, QUEUES refers to the presence (YES) or absence (NO) of interregion disk queues, and MESSAGES indicates the total number of messages queued for the region or subsystem in core and/or on overflow DDQ queues.

```
REGION          STATUS          CAUSE          MESSAGES

REGION01        ACTIVE                         10
REGION02        DOWN            NRCD           55
                  .
                  .  (up to 10 regions)
                  .
REGION10        DOWN            OPERATOR       26

{MORE        }
{NO MORE DATA}    (if last message)
```

Figure 13. Region Status Command Response

```
SUBSYS          REGION          STATUS          QUEUES      MESSAGES

H,C             REGION01        STOPPED         NO           5
H,B             REGION04        ACTIVE          NO          37
249,005         REGION04        STOPPED         YES         12
249,008         REGION06        RGN STOP*       YES          2
                                  .
                                  .  (up to 10 subsystems)
                                  .
{MORE        }
{NO MORE DATA}  (if last message)
```

*RGN STOP is a status assigned to subsystems which were active when a region was stopped. When the region is started, these subsystems will be ACTIVE without further operator action. This is differentiated from a subsystem where status is STOPPED, because a START command is required to activate these latter types.

Figure 14. Subsystem Status Command Response

Chapter 7

IMPLEMENTATION PROCEDURES

In addition to planning the configuration of the control region and satellite regions and coding multiregion tables, as previously discussed, implementation of the Multiregion Support Facility entails seven basic phases:

- Preparation of the Interregion SVC
- Preparation of multiregion modules
- Preparation of the Link Pack (RAM) area
- Preparation for use of the Dynamic Data Queuing Facility if using region-oriented disk queues
- Control region linkedit and JCL
- Satellite region linkedit and JCL
- Batch region linkedit and JCL

See also Chapter 8 of this manual regarding implementaion procedures for Region Associated Processing.

7.1 PREPARATION FOR INTERREGION COMMUNICATION

Unless already in use for VS Systems or ESS, the following steps must be performed to prepare for interregion communication:

1. Identify an available Type I SVC number.
2. Update the member SETGLOBE to indicate the Multiregion Support Facility is in use and specify the interregion SVC number:

```
&MULTREG      SETB     1       MRS IN USE

&MRSVC        SETC     'XXX'   SVC NUMBER IN DECIMAL
```

3. Resassemble and linkedit the member IGCICOM and add it to the OS/VS NUCLEUS. Linkedit parameters must be LIST, LET, REUS, DC.

7.2 PREPARATION OF MULTIREGION MODULES

The MRS modules are conditionally assembled for each operating system. Reassemble and linkedit all MRS modules that will be included in any linkedit of Intercomm: KEYFLIP, MRINTER, MRLOGOT, MRCSAMOD, MRSTAE, MROTPUT, MRPURGE, MRQMNGR, FECMD, and FESEND.

7.3 PREPARATION OF THE LINK PACK OR RAM AREA

The OS/VS Link Pack or RAM Area is used by the Multiregion Support Facility for the Multiregion Control Table (MCT). The following steps are required for MCT preparation:

1. Code, assemble and linkedit the Multiregion Communications Table (MCT). Linkedit parameters must be LIST, LET, DC, RENT.

2. Add the MCT load module to SYS1.LINKLIB (for MVS-use SYS1.LPALIB) and define the member-name (MRMCT) on SYS1.PARMLIB, using IEAIGGnn. Under MVS, MRMCT must reside in the Fixed Link Pack Area, requiring a SYS1.PARMLIB member, IEAFIXnn, which specifies MRMCT. For VS systems other than MVS, the MCT must be in the nonpageable portion of the Link Pack Area or RAM Area.

3. Verify that MRMCT does not exist in any STEPLIB or JOBLIB of an Intercomm control, satellite or batch region execution JCL.

The MCT is automatically loaded during IPL. For initial testing of the control region, the MCT need not reside in the Link Pack Area or RAM Area. See Section 7.5.

Several Intercomm modules are also eligible for the Link Pack or RAM Area, providing further savings in storage requirements for the control and satellite regions. Refer to the Operating Reference Manual for detailed implementation procedures for adding these (and user) routines to the Link Pack or RAM area, via the Intercomm Link Pack Facility.

7.4 PREPARATION FOR USE OF DYNAMIC DATA QUEUING FACILITY

The Dynamic Data Queuing Facility is used to maintain region-oriented disk queues for queue overflow processing in the control region (see Chapter 2). (See Dynamic Data Queuing Facility for detailed information.) Summary specification details are provided in Chapter 9. The following steps are required to prepare for use of the DDQ Facility.

1. Update the member DDQENV to indicate requirements via the following globals. (These settings result in reducing storage requirements.)

```
&MAXLEN     SETA     32760   (default value)
&EXTNUM     SETA     16      (default value)
&UPDAT      SETB     0       (if only MRS uses DDQ)
&WRTHEAD    SETB     0       (if only MRS uses DDQ)
&STATS      SETB     0       obsolete-required setting
&TRONLY     SETB     1       (if only MRS uses DDQ)
&SHARED     SETB     0       (if only MRS uses DDQ)
&INTLOCK    SETB     1       (required MRS setting)
```

2. Assemble and linkedit DDQMOD and DDQSTART to reflect the new DDQENV global settings.

3. Define disk data sets to be used (in the MRS control region only) by the Dynamic Data Queuing Facility with the DDQ Data Set Table (generated by DDQDS macros). Code, assemble and linkedit the table, which will then be resident in the control region.

4. All DDQ data sets must be preformatted by using the off-line Utility CREATEGF. The data set blocksize chosen should be based on average message length (including the Message Header), and whether or not the queues are blocked (specified via the DDQDS macro). The following shows typical JCL:

```
//DDQF        EXEC  PGM=CREATEGF
//STEPLIB     DD    DSN=PMI.MODREL,DISP=SHR
//MRSDDQ      DD    DSN=PMI.MRSDDQ,DISP=(,CATLG),UNIT=xxxx,
//                  VOL=SER=xxxxx,DCB=(DSORG=DA,BLKSIZE=xxxx),
//                  SPACE=......
//SYSPRINT    DD    SYSOUT=A
//SYSIN       DD    *
F MRSDDQ      nnnnn     (number of blocks)
/*
```

The number of blocks (nnnnn) to create for a DDQ data set can be estimated by multiplying the sum of QSPACE parameters for REGION and SUBSYS macros with this DDQ ddname by the value of the DDQENV global EXTNUM (default 16).

Each individual region queue is an independent logical queue. The queue space is treated in a wraparound and reusable fashion. If not enough space is allocated for that queue (via the REGION or SUBSYS macro, QSPACE parameter), a "message lost" condition occurs.

REGION macro COREQ and QSPACE parameter coding optimization can be determined by periodic examiniation of C1 and C2 log records and the buildup/time delay between them for specific subsystems or regions.

7.5 CONTROL REGION EXECUTION

The following steps are required for preparation and execution of the control region:

1. Code, assemble and linkedit the Region Descriptor Table (RDT) with member name PMIRDTnn, where nn is the operator reply at execution time to request loading of a particular RDT (see Chapter 3). (In MVS systems, the linkedit for the RDT should not specify RENT. Protection exceptions will result if this caution is not followed.)

2. Code, assemble and linkedit the control region SPA and SPA Extension (CSECTs SPA and SPAEXT) with multiregion parameters, and include SCT entries, as described previously in Chapter 3.

3. Update the Verb Table with verb chosen for the Multiregion Control Subsystem as, for example:

```
COMM BTVERB VERB=COMM,SSC=K[,SECUR=YES]
```

   If the verb chosen is not COMM, a change must be made to the module CLOSDWN3, which generates a multiregion command message with COMM as the verb in the message text.

4. Use the ICOMLINK macro to generate the control region. Specify MULTREG=CONTROL, or MULTREG=(CONTROL,1LOG) if using Single Log Feature.

5. Ensure that the following INCLUDE cards are part of the linkedit deck (lowercase letters indicate user-defined names):

```
INCLUDE SYSLIB(KEYFLIP)        Prior to IJKDSP01
INCLUDE SYSLIB(DDQMOD,DDQSTART,ddqdstbl)
INCLUDE SYSLIB(MRINTER,MRINPUT,MRQMNGR,MRSTAE,MRPURGE)
INCLUDE SYSLIB(spa,spaext)
INCLUDE SYSLIB(MRCONSS)        MRS control subsystem (not
                               included if dynamically
                               loaded)
INCLUDE SYSLIB(PMIOUTPT)       Output Utility (not included
                               if in Link Pack Area)
INCLUDE SYSLIB(MRLOGIN)        Only for Single Log feature
INCLUDE SYSLIB(MRCSAMOD)       For MVS only
```

   The following CSECTS may be inserted in overlay regions:

```
Startup Overlay A:                    MRSTART, DDQSTART
Closedown Overlay A:                  MRCLOSE
Transient Subroutine Overlay TRANS:   DDQTRANS
```

6. Ensure that the linkedit deck has INCLUDE cards for all required service routines (except those which are Link Pack resident), tables and subsystems (except those dynamically loaded).

7. Linkedit the control region load module.

8. Execution JCL must contain DD statements (in addition to the standard requirements), as follows:

   - STEPLIB: the library containing the RDT. (Must not contain MRMCT.)

   - DDQ data sets for region queues: if a data set defined in DDQDSTBL does not have a DD card or is not properly formatted, the DDQSTART module issues an indicative message and the data set is not available for queue overflow processing. (See Section 7.4.) Sample JCL:

```
//DDQname DD DSN=dsname,DISP=OLD,DCB=(DSORG=DA,OPTCD=RF)
```

9. Execute Intercomm. The control region must have the highest operating system dispatching priority for Restart/Recovery to function properly.

The NRCD and IMCD commands close down all regions. The following command closes down the control region only:

NRCD$RCONTROL@

Satellite regions then wait for the control region to become active again or abend, based upon an operator's reply to WTOR message RC004R, or SPALIST MRAUTO coding, as applicable.

Refer to Section 6.1 for the MRS control subset commands that may be entered to control individual satellite regions or subsystems.

For limited testing of the control region, the Multiregion Communications Table need not be resident in the Link Pack or RAM Area. MRSTART issues a LOAD; thus, with the proper STEPLIB DD statement, the MCT can be loaded into the control region. The user can then verify Front End tables, test the use of the Multiregion SVC, test control region subsystems, and so on. However, the user may not operate with any satellite regions. If executing in test mode, multiregion facilities may not be used, since test mode does not initialize the multiregion environment.

7.6 SATELLITE REGION EXECUTION

The following steps are required for preparation and execution of each satellite region:

1. Code, assemble and linkedit the satellite region SPA and SPA Extension (CSECTs SPA and SPAEXT) with multiregion parameters, and include SCT entries, as described in Chapter 3.

2. Use the ICOMLINK macro to generate a linkedit deck. Specify MULTREG=SATLITE, or MULTREG=(SATLITE,1LOG) if using the Single Log feature. Remove INCLUDE statements for the Output Utility if it is not defined for the satellite region (see ICOMLINK UTILITY parameter).

3. Ensure that the following INCLUDE statements are part of the linkedit deck (lowercase letters indicate user-defined names):

```
INCLUDE SYSLIB(KEYFLIP)       Prior to IJKDSP01
INCLUDE SYSLIB(MRINTER,MRINPUT,MRSTAE,MRPURGE)
INCLUDE SYSLIB(spa,spaext)
INCLUDE SYSLIB(MROTPUT)       Multiregion Output Subsystem
INCLUDE SYSLIB(MRLOGOT)       Only for Single Log Feature
INCLUDE SYSLIB(MRCSAMOD)      Only for MVS
```

   The following CSECTs may be inserted in overlay regions:

   - Startup Overlay A: MRSTART
   - Closedown Overlay A: MRCLOSE

4. Ensure that the linkedit deck has INCLUDE statements for all required service routines (unless Link pack resident), tables and subsystems (except those dynamically loaded).

5. USRSTART, if included in the linkedit, generates the 'Intercomm Started' message for the broadcast terminal group TOALL every time the satellite region is brought up. If the Output Utility is in the satellite region, the TOALL group defined in that region's PMIBROAD (broadcast table) is used. If Output is only defined in the control region SYCTTBLs, the control region's TOALL group is used, which may have undesirable results. In this case, USRSTART in each satellite region should be modified to specify a unique region-associated terminal broadcast group TID in the MSGHTID field. Eash group must then be defined in the control region's PMIBROAD module. The above discussion does not apply if the Extended Security System is in use, as USRSTART processing is bypassed.

6. Linkedit the satellite region load module.

   If the dynamic linkedit feature is used for dynamically loaded subsystems (to resolve addresses by zapping the load module at startup time), and the subsystem is used by more than one region, a separate copy of the subsystem load module must exist on a separate library for each region.

7. Execute Intercomm. There are no special execution JCL requirements for a satellite region under MRS.

8. Refer to Chapter 6 for closedown commands.

7.7 BATCH REGION EXECUTION

The following steps are required for preparation and execution of a batch region:

1. Ensure that there is a REGCOM entry in the Multiregion Communications Table with RGNID=BATCH.

2. Assemble and linkedit the member MRBATCH.

3. Include the member MRBATCH in the linkedit of each batch region that communicates with Intercomm (see also Chapter 5).

4. Execute the batch region. There are no special requirements for the execution JCL.

7.8 INTERCOMM SYSTEM WITH CONTROL REGION AND BATCH REGIONS ONLY

The installation procedure for this unique situation allows the user to bypass certain steps in preparing for control region execution. Batch region execution procedures remains unchanged. For the control region, however, the following applies:

1. No RDT is required. (Reply NO to the WTOR requesting the RDT number.)

2. MRCONSS is not required.

3. MRSTAE is optional (if batch regions are to be notified when Intercomm goes down).

7.9 RESTART/RECOVERY PROCEDURES UNDER MRS

The following summarizes the Restart/Recovery procedures available to the user for four different cases of abnormal termination:

1. Control region abend: all satellite perform SPALIST MRAUTO request processing or error message RC004R is displayed. A WAIT request places the satellite regions in a wait state until the control region is restarted; DUMP, causes the satellite regions to terminate abnormally with a dump; or ENDN, causes the satellite regions to terminate abnormally without a dump.

   Logging of messages up to the point of the abnormal termination of the control region is handled as specified in the RDT. See Section 2.3 for logging and restart options available to the user.

2. Satellite region abend without File Recovery or DBMS recovery: the control region detects that the satellite region has abended and waits for the satellite region to restart. Disposition of further messages to the satellite region depends on RDT coding and/or MRS control command FLUSH/STOP request. See also Section 2.3.1.

3. Satellite region abend with File Recovery: files must be restored from back copies in the case where the satellite region termination was due to hardware failure on the drives. Restart of the satellite region with File Recovery may then be performed. For further recommendation on file/region assignments, see Section 2.3.2. Disposition of messages to the satellite region depends on RDT coding or via MRS control command.

4. Satellite region abend with DBMS recovery: In this case, the DBMS resides in a separate region. Recovery from abend in either of these regions requires that both the DBMS region and the satellite region be restarted. The data base must be recovered and the region must be restarted. Disposition of messages that have been queued by the control region for the satellite region during restart is handled as specified in the RDT or via MRS control command. See also Section 2.3.3.

Chapter 8

REGION ASSOCIATED PROCESSING (RAP)

MRS provides isolation, security and independence for individual application subsystems that operate in separate satellite regions. In a decentralized processing environment (such as a corporate data center providing separate satellite regions for each and every user group served by the on-line system), each satellite region must effectively operate as a unique on-line system. Region Associated Processing (RAP) provides this capability.

RAP may not be selectively implemented for some satellite regions with standard MRS implemented for others. Either a standard MRS system, as previously described, or a RAP system must be chosen. Although RAP offers many extended capabilities that exceed standard MRS services, it does impose some restrictions. For this reason, it is optional.

RAP allows for single verb access to subsystems duplicated across regions, provided that the duplicated subsystem codes are identical. If editing is required, it must be done in the individual regions unless the resulting formats are identical. This may make edit before queuing impossible. Also, with RAP, message switching between subsystems in different satellite regions is not allowed. Messages may only be queued to other subsystems in the same region, or to the control region.

RAP provides for the association of a terminal to a user-specified satellite region. Consequently, each terminal can enter transactions that are processed only by subsystems in the specified satellite region or in the control region. An associated terminal is totally isolated from any subsystems in any other satellite region. Thus RAP offers the following advantages:

- RAP provides a high level of system security, since the terminal is excluded by system software from accessing programs, files, data bases or storage in other satellite regions.

- RAP prompts greater independence of satellite regions. With RAP, it is possible to duplicate subsystem codes. Any region, including the control region, may have the identical subsystem codes as any other region. (No region may have identical subsystem codes for subsystems within that region.)

- RAP allows execution of concurrent test and production satellite regions. The test region can mirror the current verb and subsystem codes of the production region. Of course, it can also have its own new verbs, subsystem codes and subsystems. This test region can operate with no danger of bringing down the production environment. By selectively

associating a terminal to either a production or test region, any terminal can be used for either production or test work. Further, RAP permits flip/flop between test and production regions through a lock/unlock command facility.

- With RAP it is possible to duplicate Intercomm utilities or service routines, such as PAGE, CHANGE/DISPLAY or GPSS, in each region without necessitating changes to operating instructions or standard installation techniques.

The following are three ways in which a terminal may be associated with a region:

1. Table Initialization

   The MRPASSW parameter on the BTERM macro for a BTAM or TCAM terminal, or on the LUNIT/LCOMP macro for VTAM, can specify the password for the associated region. In this way, beginning with startup, the terminal is permanently associated to the specified region (unless the association is changed or removed as described in the following steps). If this parameter is not specified, the terminal can access any region. A VTAM terminal can also be locked at startup, during LOGON processing, or at ESS SIGNON time via a user exit (see below). An attempt to enter a verb for a subsystem that is duplicated in two or more satellite regions (and not in the control region) will have unpredictable results if the terminal is not locked to a region. Thus, it is recommended that if duplicate subsystems are defined, all terminals should be locked to a specific region.

2. System Control Command

   Association of a terminal to a region can be defined or changed by entering a LOKR or ULKR command. These commands are processed as a Front End verb in the control region and require a password which is validated before the request is honored. Thus, terminal users lacking knowledge of requisite passwords are effectively restricted to their assigned regions.

3. Internally-Generated Command

   As with any other verb in Intercomm, the LOKR/ULKR commands can be generated by a user exit or subsystem, so that an application program itself can alter association from one region to another for a specified terminal. This generated transaction need not reassign the same terminal that triggered the subsystem: a program can reassign another terminal. Also, the Time Zone Table can be used to perform reassignment on a time-of-day basis. (Again, the program must include the password in the generated transaction, thus preserving the security aspects of RAP.)

To generate a transaction, the program sends a preformatted message to FESEND with the standard message header followed by the command format for the transaction. The receiving subsystem code should be X'0'; VMI must be X'57'. The message never gets to the terminal, but is trapped in the control region Front End and directed to a Front End routine which changes the association of terminals to regions (if the password is valid).

The implementation of RAP is fully downward compatible with the Multiregion Support Facility. Thus, if the region associator is not specified on a terminal-definition macro and the region associator program MRMOD is not INCLUDEd, RAP cannot operate. MRS then operates, as described previously, with any terminal having the capability of executing subsystems in any region. Also, the original restriction of no duplicate subsystem codes is then in effect.

8.1 IMPLEMENTATION SPECIFICATIONS

Implementation of RAP requires coding the MRPASSW parameter of applicable BTERM or LUNIT/LCOMP macros and coding the MRPASWRD macro to generate entries in the RDT. MRPASSW is defined as a one-to-eight-character alphameric field which must correspond to the value coded for the P parameter of an MRPASWRD macro in the RDT.

A one-byte field in the BTERM/LCOMP expansion is used to store the relative region number. A relative region number of zero (not locked) allows access to all regions, which requires extreme care; if duplicate subsystem codes are defined across satellite regions, which region will receive the message may be unpredictable. The control region SCT is always searched first if the terminal is not locked. If LOCKEXE=YES is coded on a BTVERB macro, RAP processing for that verb is ignored--it will execute in the control region (if defined in the SCT). Some system commands are by default lock exempt.

The RDT entries are generated by the MRPASWRD macro. This macro specifies a password and its associated region identifier. Any number of passwords can map onto a given region, but no password may map onto more than one region.

For the coding of MRPASWRD, see Appendix A. Figure 15 illustrates an example of an RDT using the MRPASWRD macro. A maximum of 100 unique passwords may be defined. For additional information on coding the RDT, refer to Chapter 3 of this manual.

8.1.1 RAP Logic

At startup, for each terminal, MRSETLOK (a routine in MRMOD) checks if MRPASSW is specified and is valid (that is, also

specified via a MRPASWRD macro). If so, the relative region number is inserted into the PTRRID field of the BTERM macro, or the LUCRID field of the LCOMP macro. In all cases, MRSETLOK will issue a message to the CPU console, the SYSPRINT data set, and the control terminal as to whether the password specified in the MRPASSW parameter is valid.

```
//GENRDT      EXEC   LIBELINK,Q=LIB,NAME=PMIRDTnn,LMOD=PMIRDTnn
//LIB.SYSIN   DD     *
./ ADD        NAME=PMIRDTnn,LIST=ALL
****          MULTIREGION DESCRIPTOR TABLE ****
****          IDENTIFIER OF REGION CONFIGURATION IS nn ****
****          CSECT IS INTERNALLY GENERATED ****
              REGION     xxxxxxxx,...
              SUBSYS     x,1,...
              SUBSYS     x,2,...
                :
              REGION     yyyyyyyy,...
              SUBSYS     y,1,...
              SUBSYS     y,2,...
                :
              MRPASWRD   P=ABCD2345,R=xxxxxxxx
              MRPASWRD   P=ABCD4567,R=yyyyyyyy
                :
              GENRDT     (No operands.  Must be the last macro.)
              END
//LKED.SYSIN  DD         *
              ENTRY      PMIRDT
              NAME       PMIRDTnn(R)
```

Figure 15. Coding an RDT using RAP

8.1.2 RAP Commands

The Front End module MRMOD processes the terminal/region associating and disassociating commands. The command verb for association is LOKR, for disassociation, ULKR.

The LOKR and ULKR commands must be defined in the Verb Table with BTVERB macros specifying no subsystem code, as these are Front End verbs. The command formats are as follows:

```
{LOKR}$password[,(tid[...,tid])]@
{ULKR}
```

where:

- LOKR associates the terminal or terminals with the region associated with the password specified.
- ULKR disassociates the terminal or terminals from the region associated with the password specified.
- password is a valid password associated with the desired region via a MRPASWRD macro.
- tid is a terminal-ID. If no terminal-ID is given, the command is assumed to refer to the terminal originating it. Valid TIDs are still processed if an invalid TID is specified.

Association of a terminal to a region may be dynamically altered at any time, that is, whether or not it is currently associated.

8.1.3 Installation Procedure

MRMOD must be included in the Intercomm control region linkedit. BTVERBs for LOKR and ULKR must be defined as follows:

```
[Symbol] BTVERB VERB={LOKR},LOCKEXE=YES[,SECUR=YES]
                     {ULKR}
```

8.2 MESSAGES ISSUED BY RAP PROCESSING

The RAP facility can associate terminals to regions at startup time and/or via RAP commands. Messages BI200I through BI203A may be issued at startup time to the control terminal and/or CPU console. If the commands are used, messages RC024I through RC036I may be returned to the issuing terminal. Message numbers are stripped from messages sent to terminals in response to the commands. Refer to these message numbers in Messages and Codes for an explanation of the messages.

8.3 MRS Control Commands with RAP

Subsystem-oriented MRS commands in the RAP environment must specify the region password in the text of the command if the command is to apply only to that region's subsystems. Otherwise all satellite regions will be checked for the requested subsystem code(s). For a specific region, the command must be entered as follows:

```
vvvv${FLUSH }$SS$P=password$...rest of text...
     {STOP  }
     {START }
     {STATUS}
```

where password is any valid password associated with the desired region via a MRPASWRD macro (see Figure 15).

For possible error responses, see Section 6.2.

Chapter 9

MRS WITH THE EDIT, PAGE AND DYNAMIC DATA QUEUING FACILITIES

9.1 MRS WITH THE EDIT UTILITY

The following considerations apply to the use of the Edit Utility in a multiregion processing environment:

1. Satellite regions which contain subsystems utilizing the edit before queuing facility: Edit Utility routines and the Edit Control Table (PMIVERBS) must be included in the control region. (If all verbs in the MRS system specify EDIT=NO or EDIT=BQ, then no satellite region need include the Edit Utility or Edit Control Table.)

2. Satellite regions containing subsystems which request editing of input messages: In BTVERB macro, specify EDIT=YES. The regions which may receive input messages to be edited must contain all appropriate Edit Utility support, that is, PMIEDIT, PMIFIXED, EDIT000 (other Edit Utility subroutines), and PMIVERBS containing all the verbs which may be edited in the region.

See the Utilities Users Guide for more information.

9.2 MRS WITH THE PAGE FACILITY

In order to use the Page Facility with MRS, the Page Facility routines must exist in each satellite region which uses the facility. A DD statement defining a unique Page data set must also be present for each region using the facility.

The following entries are required in the Front End Verb Table (BTVRBTB) for PAGE and SAVE commands:

```
BTVERB  PAGE,EDIT=BQ
BTVERB  SAVE,EDIT=BQ
```

The verbs must also be defined in the PMIVERBS in the control region.

PAGEMSG (the Page subsystem) and PAGE (the routine called by applications) share the PAGETBL CSECT (Page Table) and the related routine, SRCHPTBL. Each region which contains a subsystem which calls PAGE must also include PAGEMSG, PAGETBL and SRCHPTBL. A unique Page Table must be defined for each region.

If possible, include all the applications which invoke the Page Facility in one region (control or satellite), or code them to queue messages to a service subsystem in the control region which calls the Page Facility. If neither of these approaches are practical, the regions which contain applications using the Page Facility should be accessed by different subsets of terminals which must be dedicated to those regions. This can be easily accomplished via RAP implementation or substitute verbs (for PAGE and SAVE) can be coded (add to BTVRBTB and PMIVERBS) to point to different subsystem codes referencing the PAGEMSG subsystem in each region.

If the PAGE requests of a terminal are not sent to the region which queued the pages for that terminal, no output can be sent.

See Page Facility for more information.

9.3 MRS WITH THE DYNAMIC DATA QUEUING FACILITY

The Dynamic Data Queuing Facility provides both on-line and batch application programs with the ability to dynamically create, retrieve, and delete logical data sets (or queues) of records on a BDAM data set. This eliminates the need to define separate physical data sets for small, transient queues of application data and/or messages. Messages on DDQs may be blocked or unblocked at the user's option.

DDQs are used by MRS as region-oriented disk queue data sets. Complete documentation is contained in Dynamic Data Queuing Facility. This section presents a summary for MRS only.

The Region Descriptor Table (RDT) specifies a DDQ by name (DDNAME parameter of the REGION macro) for use as disk space for region-oriented queues. Several regions may share the same physical data set for disk queue space; the DDQ itself is a logical entity unique to each region. Each DDQ data set is defined in the Dynamic Data Queuing Facility Data Set Table (DDQDSTBL).

DDQs used for interregion and subsystem hold queues in the multiregion environment are transient queues; they are not preserved across restart because the MRS region-oriented log entries can be utilized to accomplish any required message restart for messages in queues at the time of system failure. Therefore, a QCF file is not needed for MRS use of the DDQs.

The QSPACE parameters of the REGION and SUBSYS macros specify the size of the DDQ extents allocated for a particular queue and override the BLOCK operand of the DDQDS macro. The number of DDQ extents per queue is defined in the DDQENV global member by &EXTNUM. The REGION macro defines the normal region-oriented disk queue; the SUBSYS macro defines the subystem hold queue for messages queued when the region is inactive.

To use the Dynamic Data Queuing Facility, the user must create the Data Set Table (DDQDSTBL). This table defines all the data sets on which queues may be created. The table is built by coding a DDQDS macro for each data set. The DDNAME parameters of the DDQDS and REGION macros must correspond for each MRS DDQ data set defined.

Subsystems use of DDQs may require creation of a Queue Control File (QCF) and Space Control File (SCF) as described in Dynamic Data Queuing Facility. When both a QCF and a SCF exist, if one is recreated or formatted, both must be reformatted at the same time. The number of entries in DDQDSTBL affects the SCF. If a DDQ is added, the SCF must be expanded and the QCF reformatted. The DDQDSTBL is in coordination by relative displacement with the SCF and QCF. Every region that uses both an SCF and QCF must use the same DDQDSTBL, which must have the DDQDS macros in the same order, and have defined the same parameters (except ACTIVE). If FECM DDQs are created in a satellite region, the DDQ must be shared across that satellite region and the control region.

The following are typical MRS DDQ specifications:

```
*TYPICAL ENTRIES FOR MULTIREGION DDQS
MRSDDQ1  DDQDS      DDNAME=MRSDDQ1,
                    RESTART=NO,
                    BLOCKNG=NO,
                    DEFAULT=YES

MRSDDQ2  DDQDS      DDNAME=MRSDDQ2,
                    RESTART=NO,
                    BLOCKNG=YES
```

9.3.1 MRS and the Backout-on-the-Fly Facility

Use of this File Recovery feature should be confined to one region. If used in more than one region, the THREDLOG data set must be unique in each region (different DSname) but each must be alike as only one can be defined in the DDQDSTBL. Additionally, the QCF file must be unique in each region. Shared DDQs may not be specified (except with a batch region).

Appendix A

MACROS

This appendix provides detailed parameter specifications for the following macros:

- GENRDT--Generate the Region Descriptor Table Index
- MRPASWRD--Specify RAP Passwords
- REGCOM--Define Satellite or Batch Region in the MCT
- REGION--Define Satellite Regions in the RDT
- SUBSYS--Define Subsystems in the RDT

GENRDT -- Generate the Region Descriptor Table Index

The GENRDT macro is coded following the last REGION, SUBSYS or MRPASWRD macro to cause the RDT index to be generated. This macro has no parameters.

This macro must be followed by an Assembler END statement.

The format of the GENRDT macro is as follows:

[symbol]	GENRDT

MRPASWRD -- Specify RAP Passwords

The MRPASWRD macro specifies the association between passwords and regions in the RDT for use with Region Associated Processing (RAP). Each MRPASWRD macro generates an entry in the Region Descriptor Table.

An MRPASWRD macro is coded in the following format:

[symbol]	MRPASWRD	P=password ,R=region-id

P

specifies the name of the password, defined as a one-to-eight-character alphameric field whose value should correspond to passwords associated with one or more terminals. See BTERM/LCOMP/LUNIT macro, MRPASSW parameter.

R

specifies the region identifier, a one-to-eight-character alphameric field corresponding to the region identifier specified in (a) a REGCOM macro in the MCT, (b) a REGION macro in the RDT, and (c) the MRID field in a satellite region SPALIST macro.

REGCOM -- Define Satellite or Batch Region

The REGCOM macro is used to define satellite regions and batch regions in the Multiregion Communications Table (MCT).

The format of the REGCOM macro is as follows:

[symbol]	REGCOM	RGNID=xxxxxxxx

RGNID

specifies a one-to-eight-character name used to identify a satellite region, as specified in the corresponding REGION macro. The REGCOM macro for batch regions must be coded with RGNID=BATCH and must only be coded once.

The control region must not be identified by a REGCOM macro; no REGCOM macro may specify RGNID=CONTROL.

REGION -- Define Satellite Regions

The REGION macro is used to define satellite regions in the Region Descriptor Table (RDT). One REGION macro must be coded for each satellite region and must be followed by one or more SUBSYS macros.

The format of the REGION macro is as follows:

[symbol]	REGION	xxxxxxxx [,BLOCKED={NO }] [{YES}] [,COREQ={nn}] [{4 }] [,CSALEN={nnnn}] [{1024}] [,DDNAME=dddddddd] [,LOG={NO }] [{YES}] [,QSPACE={nn}] [{8 }] [,STOP={YES}] [{NO }]

xxxxxxxx
: specifies the one-to-eight character name used to identify the satellite region. A corresponding RGNID value must be coded for a REGCOM macro in the MCT, and for MRID in the SPALIST for the referenced satellite region.

BLOCKED
: specifies whether the disk queue data set is to be blocked. This parameter is valid only if DDNAME is coded. Code NO if the disk queue is unblocked. The default is YES. The queueing data set defined in the DDQDSTBL must define blocked queues if BLOCKED=YES.

COREQ
: defines the maximum number of messages that may be queued in core in the control region for this satellite region. Code as a decimal value 0 to 8000. The default is 4. If 0 is coded, no messages are queued in core; however, the DDNAME parameter must then be coded to provide a disk queue.

CSALEN

specifies the amount of CSA to acquire when the satellite region becomes active and to hold until the satellite region terminates. The default is 1024. For MVS only.

DDNAME

specifies the name of the disk queue, if any, for the region. If no DDNAME is coded, there is no disk queue for the region. The DDNAME coded must correspond to an entry in the Dynamic Data Queue Data Set Table (DDQDSTBL), and must be a valid OS/VS DDNAME. Refer to Section 9.3 for a summary of coding procedures for DDQDSTBL. See also the QSPACE parameter.

LOG

specifies whether interregion message traffic from the control region to this region is to be logged on the control region's logging data set. The default is YES, providing for region queue recovery if the control region abends and is restarted; subsystem logging is governed by the logging specifications in the SUBSYS macros. If NO is coded, no interregion logging takes place. (Refer to Section 2.3.)

QSPACE

defines the number of continguous physical blocks for allocation as one DDQ extent. The DDQ Facility allocates an extent dynamically; the maximum number of extents per DDQ is defined by a global. (Refer to Section 7.4.) Code as a decimal value, 1 to 32760. The default is 8 blocks. This parameter is valid only if DDNAME is coded.

STOP

specifies the disposition of messages when the satellite region is initiated. If YES, the region does not receive input until a multiregion START command is issued. The default is NO, specifying that the region receives input as soon as it is initiated. If YES, this parameter overrides the SUBSYS macro ALT, IFDOWN and STOP parameters until the START command for this region is issued. Applies only to first startup of the region. Subsequent restarts are controlled by the MRS STOP command.

SUBSYS -- Define Subsystems

The SUBSYS macro defines subsystems in satellite regions for the RDT. There must be one SUBSYS macro for each subsystem that can receive messages from terminals (via the control region) or from the other satellite regions. A maximum of 1000 SUBSYS macros may be coded in one RDT.

The format of the SUBSYS macro follows:

[symbol]	SUBSYS	xxx,yyy [,ALT=aaaaaaaa] [,IFDOWN={FLUSH}] [{QUEUE}] [,LOG={NO }] [{YES}] [,LSYNCH={YES}] [{NO }] [,QSPACE={nn}] [{8 }] [,RESTART={NO }] [{YES}] [,STOP={YES}] [{NO }]

xxx

specifies the high-order byte of the subsystem code. Code as either a single alphameric character or as a three-digit decimal number not greater than 255 for conversion to a one-byte hexadecimal value.

yyy

specifies the low-order byte of the subsystem code. Code as either a single alphameric character or as a three-digit decimal number not greater than 255 for conversion to a one-byte hexadecimal value.

ALT

specifies the region identifier of the satellite region to which all messages for this subsystem are routed if this region is inactive. (If the alternate is also inactive, message disposition is controlled by the IFDOWN parameter.) This parameter is optional.

IFDOWN

specifies action taken by the control region regarding messages for this subsystem when the satellite region containing this subsystem (and the subsystem's alternate region, if any) is inactive. If FLUSH is coded, current and future messages queued for this subsystem are flushed (until the region is restarted). If QUEUE, the default, is specified, currently queued messages are retained and new messages for this subsystem are queued (may be dynamically overridden by the MRS FLUSH command). For IFDOWN=QUEUE, either the primary or alternate associated REGION macro must specify the DDNAME parameter; otherwise, messages are flushed. If QUEUE is specified, QSPACE must be coded.

LOG

specifies whether or not interregion message traffic from the control region to this subsystem is to be logged on the control region's logging data set. Code NO if there is no logging. The default is YES. If LOG=YES, YES must also be specified on the associated REGION macro LOG parameter.

LSYNCH

specifies whether or not interregion message traffic log entries for this subsystem are critical; that is, whether or not they must be physically written to INTERLOG before continuing processing. Code YES to specify an immediate write (synchronous logging). NO, the default, indicates that the message is queued for a write to INTERLOG (asynchronous logging).

QSPACE

specifies the number of physical blocks per DDQ extent for allocation for a subsystem hold queue to contain messages for this subsystem when the primary and alternate (if any) regions are inactive. Code as a decimal value 1 to 32760. The default is 8. This parameter is required if IFDOWN=QUEUE.

RESTART

specifies whether or not interregion message traffic log entries for this subsystem are to be restarted in case of control region restart. YES, the default, indicates restart is required (LOG=YES must be coded). NO indicates no restart will be performed.

STOP

specifies the disposition of messages when the satellite region is initiated. If YES is coded, messages directed to this subsystem remain queued until the subsystem is started for input by the MRS START command. The default is NO, specifying that the subsystem is to receive its input as soon as its associated region is started. Applies only to first startup of the region, subsequent restarts are controlled by the MRS STOP command. See also REGION macro STOP parameter.