

INTERCOMM

PLANNING GUIDE

LICENSE: INTERCOMM TELEPROCESSING MONITOR

Copyright (c) 2005, 2022, Tetragon LLC

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Use or redistribution in any form, including derivative works, must be for non-commercial purposes only.
2. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
3. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Planning Guide

Publishing History

<u>Publication</u>	<u>Date</u>	<u>Remarks</u>
First Edition	September 1973	This manual corresponds to Intercomm Release 6.0.
Second Edition	March 1976	Corresponding to Intercomm Release 7.0.
Third Edition	September 1978	Corresponding to Intercomm Release 8.0.
Fourth Edition	December 1982	Corresponding to Intercomm Release 9.0.
Fifth Edition	April 1988	Corresponding to Intercomm Release 10.
Sixth Edition	January 1995	Additions for Intercomm Release 10 through SM level 2300.

The material in this document is proprietary and confidential. Any reproduction of this material without the written permission of Isogon Corporation is prohibited.

PREFACE

Intercomm is a state-of-the-art teleprocessing monitor system executing on the IBM System/370 and System/390 family of computers and operating under the control of IBM Operating Systems (XA and ESA). Intercomm monitors the transmission of messages to and 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.

The Intercomm product is under continual development designed to keep the system at the forefront of computer technology. This document describes the incremental enhancements that have been added to our latest release, Release 10. This release represents a major technology upgrade, incorporating many new features that improve system reliability, ease on-line programming requirements, and facilitate maintenance and installation.

This manual provides a synopsis of each of the Release 10 features. While every attempt has been made to ensure the technical accuracy of data in this document, the reader should refer to the various system technical documents for feature implementation specifics.

It is assumed the reader has a basic familiarity with the Intercomm Teleprocessing Monitor System, its features and tables.

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

Model System Generator

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

	<u>Page</u>
Chapter 1 INTRODUCTION	1-1
Chapter 2 RELEASE 10 FEATURES	2-1
2.1 Feature Categories	2-1
2.2 General System Enhancements	2-2
2.2.1 Message Header Revisions	2-3
2.2.2 Optionally Load BTVRBTB at Startup	2-4
2.2.3 Link Pack Facility Installation	2-5
2.2.4 Procs Revised to Use Assembler H	2-6
2.2.4.1 New VS COBOL II Procs	2-6
2.2.5 In-Core Table Sort Service Routine	2-7
2.2.6 Support Large 3270 Buffers for Output Utility	2-7
2.2.7 Intercomm Integrity SVC (IISVC)	2-8
2.2.8 Enhanced LU6.2 Support	2-9
2.2.9 31-Amode Dynamic Storage Support	2-10
2.2.10 Table Facility	2-11
2.2.11 Optimized Page Facility	2-12
2.2.12 31-Amode Intercomm Pools	2-13
2.3 General System Improvements	2-15
2.4 User Exit Changes	2-18
2.4.1 COPYEXIT User Exit	2-19
2.4.2 USROTEDT - FESEND Call	2-19
2.4.3 USERB37E User Exit	2-20
2.4.4 PREPROGI and PREPROGE Exits	2-20
2.4.5 USRPAGEX User Exit	2-21
2.4.6 USRQMONX User Exit	2-21
2.4.7 MRSECUR2 User Exit	2-22
2.4.8 SECUEXIT and LUCUR ESS Exits	2-22
2.4.9 USRSTSCH User Exit	2-22
2.5 Desupported Facilities	2-23
2.6 Program Processing Optimizations	2-24
2.6.1 Subsystem Initialization and Control	2-26
2.6.2 Subsystem Disk Queuing Always FIFO	2-26
2.6.3 Storage Availability Checking Eliminated	2-27
2.6.4 Additional Subsystem Processing Parameters	2-28
2.6.5 Indicative Dump Control by ID and Subsystem	2-29
2.6.6 COBOL Subsystem GET/FREE Optimization	2-29
2.6.7 Dynamic Subsystem Loading Optimization	2-30
2.6.8 Dynamic Subroutine Loading Optimization	2-30
2.6.9 COBOL Program Loading Above the 16M Line	2-31
2.6.9.1 VS COBOL II Support	2-31
2.6.10 FINTUNER Command Revisions	2-32
2.6.11 'Program Cancelled' Message Improvements	2-33
2.6.12 PMISERC3 Utility Subroutine Optimization	2-33
2.6.13 PMIDCB Module Changes	2-34
2.6.14 FASTSNAP Processing Changes	2-34
2.6.15 Automated Message Restart Facility	2-35
2.6.16 'Hung' Thread Recovery	2-36
2.2.17 SAM Enhancements	2-37

		<u>Page</u>
2.6.18	Subsystem Parameter List Accessibility	2-38
2.6.19	Save Area Chaining	2-39
2.6.20	Assembler Program Loading Above the 16M Line	2-40
2.6.21	PL/1 Program Loading Above the 16M Line	2-41
2.6.22	Direct Calls to User Subroutine Psuedo-Entries in INTLOAD	2-42
2.6.23	Continuous Multithreading of Subsystems Loaded Above the 16M Line	2-43
2.7	Statistics Gathering and System Display Upgrades	2-44
2.7.1	New System Control/Display Command - SCTL	2-46
2.8	BTAM/TCAM Front End Changes	2-47
2.9	VTAM Front End Changes	2-48
2.10	Message Mapping Utilities (MMU) Changes	2-50
2.11	VSAM Support Enhancements	2-51
2.11.1	ESDS File Loading Support	2-52
2.11.2	Path Data Set Name Sharing Support	2-52
2.11.3	New LSR Buffer Pool Sizes	2-53
2.11.4	LSR Pool Above the 16M Line	2-53
2.12	Dynamic File Allocation Special Feature Modifications	2-54
2.13	TOTAL Data Base Support Enhancements	2-55
2.14	Extended Security System Enhancements	2-56
2.15	System Control Command Enhancements	2-58
Chapter 3	NEW DOCUMENTATION	3-1
Appendix A	Intercomm Modules Added/Deleted/Rewritten for Release 10	A-1
Appendix B	New Intercomm Message Header Layout	B-1

Chapter 1

INTRODUCTION

Release 10 of Intercomm contains extensive and significant improvements, affecting many areas of the Intercomm teleprocessing monitor. Among the Release 10 features are:

- Implementation of many Intercomm User Group PER requests
- Addition of written and telephoned requests from users
- Incorporation of funded development requests
- Simplified table coding/expansion procedures
- Debugging aids
- Performance and reliability improvements
- New options and features
- Eased restrictions
- New on-line system maintenance and display commands
- Virtual storage constraint relief

Testing procedures for Release 10 included both extensive in-house testing and field Beta testing at several user sites which were chosen for the variety of options to be tested and representative operating systems in use. The major documentation updates describing the installation and use of the enhancements were completed prior to Beta testing. Thus, both the Release 10 documentation and the code could be field-tested and corrected where necessary before official release to the user base. As a result of this procedure, users are assured of a stable release, and that upgrading to Release 10 can be accomplished smoothly.

In the following chapter, the Release 10 enhancements are described in detail, along with the accompanying documentation upgrades, and conversion considerations for current users of Intercomm.

All of the enhancements are in production at one or more Release 10 sites. For example, at least 5 sites are known to be using the new IISVC (Intercomm Integrity SVC), and 1 site has already converted to VS COBOL II, while another is in the process of conversion.



Chapter 2

RELEASE 10 FEATURES

2.1 FEATURE CATEGORIES

For convenience and planning purposes, the Release 10 enhancements and new features have been grouped into the following categories:

- General system enhancements
- General system improvements
- User Exit changes
- Desupported facilities
- User program processing optimizations
- Statistics gathering and system display upgrades
- BTAM/TCAM Front End changes
- VTAM Front End changes
- Message Mapping Utilities changes
- VSAM support changes
- Dynamic File Allocation special feature rewrite
- TOTAL Data Base support enhancements
- Extended Security System (ESS) enhancements
- System control command changes and additions.

For each of these categories, the individual items will be described together with the impact, if any, on system externals. A few of these improvements were originally available and tested as Experimental SMs (and later provided as Release 9 SMs above the 1730 level). However, the interrelationship with other enhancements necessitated incorporation and modification of such code into official Release 10.

2.2 GENERAL SYSTEM ENHANCEMENTS

System enhancements described in detail in the following sections are:

- MSGHBMN field expanded in Message Header to 3 bytes and some fields moved/reorganized
- The BTVRBTB may be loaded at Startup
- Link Pack Facility installation provided
- Procs using Assembler upgraded to Assembler H
- Procs added for VS COBOL II
- In-core Table Sort service routine (INTSORT)
- Output Utility supports large 3270 buffers
- Generic table entries may be used in the Station Table (see Section 2.4)
- Loading above 16M line supported for user subsystems/subroutines (see Section 2.6)
- Intercomm Integrity SVC (IISVC global) added: use instead of MRSVC if MVS Operating System access security desired
- LU6.2 external feature enhanced to add active mode (Intercomm to another APPLID) support as an extra cost feature
- VS COBOL II support provided (see Section 2.6)
- STORAGE/STORFREE/PASS/CATCH macros support 31-Amode storage requests
- New Table Facility developed for user tables in 31-Amode storage
- Page Facility redesigned to use Table Facility
- 31-Amode Intercomm pools support (post SM level 2300).

2.2.1 Message Header Revisions

For high-volume systems and/or for systems which execute continuously for several days, the limitation of a 2-byte field for the Front End input message sequence number can cause a wrap-around of that number (MSGHBMN). A wrap-around (from 64K-1 to 1) causes Log Analysis problems, in addition to causing confusion when analyzing a LOGPRINT printout. Therefore, the obsolete 5-byte MSGHPID field has been modified so that the last 3 bytes now contain the new BMN number field. The old 2-byte BMN number field is now used for internal Front End processing. See new Message Header layout in Appendix B.

All Intercomm modules which reference or display the BMN field have been modified accordingly, including the Log printing and analysis modules. Additionally, the Message Header COPY members for user programs have been updated. The user must check all user exit routines, application subsystems and subroutines, and batch programs which reference the BMN field to determine if corresponding coding changes are necessary.

In addition, the MSGHCON field has been eliminated (STATION macro Company Numbers obsolete) and replaced with the 2-byte MSGHFLGS field (Front End message flags) from the first 2 bytes of the old MSGHPID field. Internal flag settings formerly in the MSGHCON field have been merged into the replacement MSGHFLGS area: this merge cannot be backed off. User programs referencing MSGHCON and its values must be modified. User programs referencing MSGHFLGS (MSGHFLG1 and MSGHFLG2) must be reassembled. Also note that the MSGHBLK field is reserved and used by the Front End; it is also used by the Back End to hold the MSGHMRDX index for input messages to a Satellite Region from the Control Region in a Multiregion environment and to hold the MSGHRETN (return code) for the FA log record on subsystem thread completion. MSGHMRDX and MSGHRETN previously overlaid the first and second bytes of MSGHCON, respectively. Reassemble user programs that reference those fields.

Copy member MSGHDRC has also been reorganized to put the File Recovery message header fields in a separate layout at the bottom and to indicate which fields are the same as in a standard message header.

NOTE: a change deck is available to back off the BMN field change (use old 2-byte field) if necessary.

Documentation Changes: All manuals illustrating the Message Header including:
[Operating Reference Manual](#)
[Programmers Guides](#)
[File Recovery Users Guide](#)

New External Options: None.

Installation: Automatic (review user programs referencing revised MSGHDR fields).
Copy language-dependent message header COPY members to user copy libraries.

2.2.2 Optionally Load BTVRBTB at Startup

If the BTVRBTB macros defining input message transaction codes (verbs) are coded in a separate module (BTVRBTB) from the terminal definition macros, then the BTVRBTB can be loaded at startup instead of being included in the Intercomm load module. This means that if verbs are being added to the table during a development phase, it is not necessary to relink Intercomm to incorporate the revised BTVRBTB. The BTVRBTB load module must be relinked as BTVRBTBnn where nn is a 2-digit suffix from 00 to 99. Thus, different versions may be defined: the version to be used is requested at startup. The linkedit of BTVRBTBnn must have an ENTRY VERBVCON statement to enable entry points resolution after loading.

At startup, if the BTVRBTB is not found in the Intercomm load module (Control Region) but the loading module VERBSTRT is, then a WTOR message (MI080R) is issued requesting the suffix number of the BTVRBTBs table to be loaded. If the BTVRBTBnn module is not found on STEPLIB, another message is issued with the options to retry the load (message MI080R reissued), or to cancel or abend (U0199) Intercomm. If successfully loaded, the addresses of the BTVRBTB table and its index are resolved into the SPA (SPAVRBTB and SPAVBNDX) for future reference by Intercomm modules. User routines which used to use a V(BTVRBTB) to reference the verb table must be changed to reference the SPA fields when the BTVRBTB is dynamically loaded. Fields in the BTSPA which reference the table are also resolved when the table is loaded.

Documentation Changes: Operating Reference Manual
 Installation Guide
 Basic System Macros

New External Options: ICOMGEN Macro: DYNVERB parameter.
 ICOMLINK Macro: DYNVERB parameter.

Installation: Include VERBSTRT with startup modules, omit BTVRBTB
 from linkedit. Rename BTVRBTB load module to
 BTVRBTBnn (Csect name must be BTVRBTB).

2.2.3 Link Pack Facility Installation

If a Multiregion Intercomm system, or multiple Intercomm jobs, are executed on one CPU, a Link Pack Facility is provided to place over 100K of reentrant Intercomm modules in the system Pageable Link Pack Area (PLPA). This enhancement provides for:

- Generation of Installation JCL,
- Automatic revision of generated linkedit statements.

Depending on processing options chosen for the installation macro ICOMGEN, and coding LPSPA=YES, jobs are generated to relink link pack eligible Intercomm modules as reentrant, to generate LPSPA and LPINTFC macro coding and then to assemble the macros, and to link the LPSPA load module to the system link library. Additionally, coding LPSPA=YES on the ICOMLINK macro will cause omission of link pack eligible modules from the generated INCLUDE statements, and generate INCLUDEs for the assembled LPINTFC table and the LPSTART startup resolution module. Note that the MMU control command subsystems (MMUCOMM and LMAP) are also eligible and processed accordingly.

Documentation Changes: Basic System Macros
 Installation Guide
 Operating Reference Manual

New External Options: ICOMGEN Macro: LPSPA parameter.
 ICOMLINK Macro: LPSPA parameter.

Installation: (see documentation) Also, add LPSPALIB DD statement to execution JCL for Csect name searches (see Operating Reference Manual).

2.2.4 Procs Revised to Use Assembler H

The following Intercomm-supplied JCL procedures which execute the Assembler have been revised for executing Assembler H (IEV90) instead of Assembler F:

ASMPCL, ASMOC, ASMPCL, LIBEASM, LIBELINK, DEFSYM, SYMGEN

This is required for XA and ESA. See the COPYPROX job in Chapter 2 of the Installation Guide for unloading these procs from SYMREL to the Intercomm or system PROCLIB.

Additionally, region size specifications were increased to 256K (adjust for site standards). A DECK parm (default=NODECK) is available for producing punched output from ASMPCL for the ICOMGEN and ICOMLINK macros (add a SYSPUNCH DD statement). The MVS library SYS1.AMODGEN is added to the SYSLIB concatenation (after SYS1.MACLIB) for all the procs.

Documentation Changes: Installation Guide
Operating Reference Manual

New External Options: ASMPCL: DECK parameter.

Installation: None.

2.2.4.1 New VS COBOL II Procs

Two procs have been added to SYMREL (and COPYPROX job) to provide for VS COBOL II support, as follows:

- COB2PC -- compile a VS COBOL II program
- COB2PCL -- compile and link (as reentrant) a VS COBOL II program: parms AM and RM added to specify AMODE and RMODE for loading above or below the 16M line (default is below - required if program is resident).

Documentation Changes: COBOL Programmers Guide
Installation Guide
Operating Reference Manual

New External Options: See COBOL Programmers Guide Appendix A for a detailed listing of these procs.

Installation: Copy procs to user proc library.

2.2.5 In-Core Table Sort Service Routine

A new service routine called INTSORT, which was developed for ESS (Extended Security System) enhancements, has been generalized for use by user programs. An efficient sort algorithm (modified bubble sort) is used. To execute INTSORT from an Assembler Language program, the following parameters are required:

- 1) Entries - fullword containing number of table entries (unlimited)
- 2) Entry-length - fullword containing table entry size (up to 32767)
- 3) Table - address of table area to be sorted
- 4) Key-offset - fullword containing entry key offset (relative to 0)
- 5) Key-length - fullword containing key length (up to 256)

To execute INTSORT from a COBOL or PL/1 program, the entry INTSORTC must be called with the above parameter list, plus a sixth parameter of the address of a fullword to contain the return code in binary in the low-order byte (returned to an Assembler program via R15) - see documentation. The name INTSORTC has been added to REENTSBS, and the ICOMSBS, PENTRY, and PLIENTRY copy members

INTSORT (INTSORTC) may also be used by batch programs (link with user module), and can be used for 31-Amode tables (used by several Intercomm modules).

Documentation Changes: Programmers Guides

New External Options: None.

Installation: Automatic (INTSORT automatically included by ICOMLINK).

2.2.6 Support Large 3270 Buffers for Output Utility

On the REPORT macro to define a user Report (RPTnnnnn module) for Output Utility processing, if DEV=3270 is coded, it was assumed that a 1920-character buffer size is to be used. However, for large screen CRTs and 132-column printers, such a limitation is undesirable. This enhancement provides two new REPORT macro parameters, ROW and COL, to define the number of rows and columns in the buffer of the destination terminals for the report. These parameters are only valid if DEV=3270 is specified. The defaults assumed are 24 rows by 80 columns if the new parameters are not coded. These parameters allow additional lines and item positions to be defined up to the limits set by the parameters.

Documentation Changes: Utility Users Guide

New External Options: REPORT macro: ROW, COL parameters.

Installation: Define large buffer size in Device Table (PMIDEVTB).

2.2.7 Intercomm Integrity SVC (IISVC)

To remove the possibility of Intercomm executing in Supervisor State and/or Protect Key 0 in open code via an SVC (MRSVC), and to replace that SVC, the Intercomm Integrity SVC was developed to prevent illegal access to the Operating System and its protected Control Blocks by non-system (Intercomm) user programs and exit routines.

The SVC (IGCICSVC) is provided only in object-code form and is a Type 2 reentrant SVC which requires the LOCAL Lock at entry. The IGCICSVC load module must be linked to SYS1.NUCLEUS and defined to the Operating System (see Installation Guide), after defining the assigned SVC number to Intercomm via the new IISVC global in SETGLOBE (also deactivate - set to '013' - the MRSVC global to guarantee integrity). Functions previously performed in Intercomm using the MRSVC have been replicated in the new SVC, which also contains code to validate the calling user and the function requested. Non-Intercomm (batch, etc.) regions may not execute IGCICSVC. The SVC also prevents accidental or intentional modification of protected storage/control blocks not previously acquired by the SVC on behalf of the same address space, and prevents modification of Link Pack Area modules via the SVC (except when used as Intercomm control blocks for Multiregion and ESS).

Storage protection is not guaranteed if the MRSVC is also implemented (for user programs) in the same Intercomm address space. Intercomm modules to reassemble when the IISVC is implemented in SETGLOBE are listed in the Installation Guide.

Documentation Changes:	<u>Installation Guide</u> <u>Operating Reference Manual</u>
New External Options:	IISVC global in INTGLOBE/SETGLOBE. IGCICSVC load module (executable SVC).
Installation:	Automatic via ICOMGEN macro (if IISVC instead of MRSVC implemented in SETGLOBE).

2.2.8 Enhanced LU6.2 Support

Enhanced (Extended) VTAM LU6.2 support, a new Intercomm external (extra cost) feature, provides full interconnectivity with other applications using the SNA LU6.2 protocol (also known as APPC). With Enhanced LU6.2 support implemented, Intercomm can both receive and initiate transactions on LU6.2 links with other VTAM applications.

Enhanced, or Extended, LU6.2 support is also referred to as Active Mode support vs. the original Basic (Passive Mode) implementation of LU6.2 session support and prerequisites, or corequisites, purchase of the Basic LU6.2 support external feature.

Active Mode support has two features for initiating a transaction to another VTAM APPLID (another Intercomm, CICS, etc.) on an LU6.2 link:

- 1) Pass Through: - verbs (transaction-ids) can be defined (in BTVRBTB) as locked to another VTAM APPLID, and transactions input to Intercomm with such a verb will be automatically passed to the specified system for processing. The response from the other APPLID will be automatically routed back to the originating 'terminal'. Thus, Intercomm can be a Front End (terminal-owning region) to other systems (and/or other Intercomms) via LU6.2 links. Dynamic locking/unlocking of a verb is provided by new LOKA and ULKA commands (see Section 2.15).
- 2) Back End: - a transaction for an LU6.2 link can be initiated by an application subsystem via a call to INITLU6 which will format the transaction request into a new FECM type (FECMLU6) and pass it to the Intercomm Front End via FESEND (MROTPUT in a Satellite Region) for queuing to the specified LU6.2 link (APPLID). The response from the other APPLID will be routed back to the calling subsystem (in a program-provided area) via INITLU6 (application thread suspended in a WAIT-LU6 state), and/or to the original input (to the subsystem) 'terminal', or to a specified other terminal (printer output), depending on INITLU6 call options.

Documentation Changes: [SNA LU6.2 Support Guide](#)
[Installation Guide](#)
[Programmers Guides](#)
[Basic System Macros](#)
[System Control Commands](#)

New External Options: BTVERB macro, APPLID parameter.
 ICOMGEN/ICOMLINK macros, LU62=ACTIVE.
 LOKA/ULKA system control commands.
 INITLU6 service routine.
 INITLU6 definitions in REENTSBS, ICOMSBS, PENTRY,
 PLIENTRY, INTLOAD, and SPA.

Installation: Automatic via ICOMGEN (see documentation).
 Ensure INITLU6 in applicable Intercomm region linkedits.
 Add LU6.2 link definitions to Front End Network Table.
 Code APPLID parameter on applicable BTVERB macros,
 as desired, and ensure new Front End verbs LOKA and
 ULKA are defined in BTVRBTB.

2.2.9 31-Amode Dynamic Storage Support

To provide for Intercomm and user program requests for dynamic storage above the 16M line, and to have those requests tracked via RCBs, a new parameter for the STORAGE macro - LOC=ANY/BELOW - has been provided. Coding LOC=ANY will cause a 31-Amode (31-bit) storage address to be returned. To address the returned storage, the calling program must then execute in 31-Amode (use XASWITCH macro to flip modes if residing in (or loaded into) 24-Amode memory, and be sure to flip back to 24-Amode before issuing other macros or calling other programs/service routines). If LOC=ANY, the LIST parameter must be used and must point to 24-Amode storage, as must also the ADDR parameter (if used).

The STORFREE, PASS, and CATCH macro entry-points, to the Resource Management module MANAGER, have also been modified to process 31-Amode addresses. The issuing user must ensure that the high-order byte of a 24-bit address passed to these entry-points is zero. The entry-points will first search RCBs for 31-bit core addresses if the high-order byte of the address is non-zero. PASS and CATCH entry-points will then do standard 24-bit address processing (after first clearing the high-order byte of the passed address). STORFREE will issue a message (RM028I) and force a program check if no match for the address with a non-zero high-order byte is found - it will not revert to standard 24-bit processing.

This support requires that RCB (RCB Table) support is implemented (see Operating Reference Manual) to ensure correct processing of 31-bit addresses.

LOC=ANY (for 31-bit dynamic storage addresses via GETMAIN) is also supported in batch mode processing for calling programs linked with the Intercomm BATCHPAK interface routine.

Documentation Changes: Basic System Macros
 Operating Reference Manual

New External Options: STORAGE macro, LOC=ANY.

Installation: &RM global must be 1 in SETGLOBE when MANAGER assembled (required).
 Link batch program with BATCHPAK to use STORAGE macro with LOC=ANY.

2.2.10 Table Facility

A new facility (INTTABLE module) has been developed for all Release 10 users (not a special feature), to provide for user and system processing of dynamic tables in 31-Amode storage. The tables must have unique names and are built in memory above the 16M line in an online or batch mode address space.

A table can be used as a scratch pad area or to hold data records. Tables can be kept and modified or reused during the life of one Intercomm or Batch Mode job step execution (not preserved/recreated across a restart), or kept only during one thread's execution.

The maximum table size is constrained only by storage availability above the 16M line. Table entries are fixed in length (specified at table creation up to 32767 bytes) and can have keys for random access after sorting by key. Sequential, reverse sequential, and random access by entry number is also supported. Entries can be added, deleted, or updated (by the same or another thread), after a table is created. For example, a table can be used to hold pertinent data record or statistical information for subsequent report printing by the same (saves 24-Amode storage/DWS/ISA space), or another, thread. Or data/statistics can be gathered for online display. The user can create variable-length entries by embedding the real entry length within the fixed-length entry area, which must be of maximum possible entry size for the table, and thus process the entry area as for variable-length data records. Tables are internally expanded as entries are added, if needed.

New entry-points for Table Facility calls from 24- or 31-Amode programs are provided. Programs retrieve/add/update entry copies, they do not process the tables themselves. Entry-points are also provided for creating (building) a table, sorting it (if entries have keys), reopening it, and deleting or keeping it. Snaps are available for tracing/debugging access calls. RMPURGE has been updated to purge tables built but not yet kept or deleted by a thread, and to close tables opened by the thread, if any. Statistics on Table Facility usage are provided (see Section 2.7), and calls to the entry-points can be tracked via SAM implementation (see Section 2.6).

Documentation Changes: Table Facility (new)
 Basic System Macros
 Programmers Guides
 Operating Reference Manual

New External Options: Table Facility entry-points added to REENTSBS, ICOMSBS, PENTRY, PLIENTRY, INTLOAD, and the SPAEXT.
 SPALIST parameters to define minimum table area sizes and increments, etc. are described in the documentation.
 STRT/STOP commands, TABSNAP option, to activate or suppress (default) table access snaps.

Installation: Automatic via ICOMLINK (ensure INTTABLE and INTSORT in linkedit).
 Code SPALIST parameters if defaults not desired.
 Implementation of STORAGE macro, LOC=ANY option for 31-Amode core required (see previous Section).
 Include INTTABLE and INTSORT with BATCHPAK for batch mode program use, if desired.

2.2.11 Optimized Page Facility

The Page Facility (CRT page browsing) has been redesigned for Release 10 to eliminate both the user-coded Page Table (PAGETBLE) and the BDAM Page Data Sets formerly used to store the pages (output messages) for later retrieval. Instead, the new Table Facility (see previous Section) is used to store the pages in terminal-oriented Response Tables above the 16M line. Thus, response time for storage and retrieval of pages is dramatically improved; DASD I/O is no longer needed. A Page Master Table is also created and maintained internally to contain terminal/response control blocks formerly in the user-coded PAGETBLE.

This redesign is transparent to user programs using the Page Facility, either by direct calls to PAGE or via the MAPEND P (put output message(s) in a Response) option. Return code meanings have been modified to reflect Table Facility access errors. Page Facility terminal operator commands are also unaffected. A new PAGE command option E allows the operator to view the last page of all Responses accumulated for the terminal (vs. option L for displaying the last page of the current Response being viewed).

By default, all input terminals are eligible for Page Facility processing. A USRPAGEX user exit (from PAGE) is provided (may be modified) to control adding pages to an existing Response and/or creation of a Response for the requesting terminal (see Section 2.4).

Documentation Changes: Page Facility (1994 Edition)
Table Facility
Basic System Macros

New External Options: SPALIST parameters for defining the maximum number of pages (messages) for any terminal across its Responses (PFMAXP), the maximum number of responses per terminal (PFMAXR), and the maximum acceptable page (message including header) size (PFMAXS).
 SCTL\$SPA\$PFMAXS command to dynamically modify the maximum page size.
 MAPACCT macro, PAGECLS option, to track calls to PAGE via SAM statistics.

Installation: Automatic via ICOMLINK (ensure PAGE, PAGEMSG, INTTABLE and INTSORT in linkedit).
 Include/remove USRPAGEX user exit, as desired.
 Ensure PAGEMSG subsystem defined in SCT and PMIRDT (if Multiregion).
 Ensure BTVRBTB macros for PAGE and SAVE in BTVRBTB.
 Ensure Edit Utility entries for PAGE, SAVE, in PMIVERBS.
 Use Table Facility statistics to tune table sizes needed for Page Facility Response Tables.

2.2.12 31-Amode Intercomm Pools

At the next SM level (after 2300) support for installation of an Intercomm 31-Amode pools module (in addition to the existing 24-Amode pools) will be available (support can be pre-installed via Experimental SMs). 31-Amode pools can be used to satisfy STORAGE requests with LOC=ANY (described in Section 2.2.9) which do not force a GETMAIN (via BOUND=PAGE, the SP (non-zero), or POOLS=NO parameters).

A 31-Amode pools module is coded using the COREACCT macro (see Section 2.7) for tuning statistics, and ICOMPOOL macros to define the number and size of the desired pool blocks. The load module name must be IC31PLnn, which is loaded at startup if POOLSTRT is included in the Intercomm linkedit. IC31PLnn must be prelinked with the RMODE=ANY and AMODE=31 parameters to force loading above the 16M line. To define the load module suffix (nn), a new parameter on the SPALIST macro (IC31PL=nn/NO) must be coded. If the IC31PL parameter is not NO, POOLSTRT attempts to load the IC31PL module with the coded 2-byte alphanumeric suffix and issues a message (WTOR) if unsuccessful/load module not 31-Amode, or resolves entry-points into the SPAEXT if successful.

MANAGER has been completely rewritten to also process the 31-Amode pools (if loaded successfully). Because 31-Amode storage request processing requires RCB's (see Section 2.2.9), the RM global has been removed from INTGLOBE and SETGLOBE and the RCBTABLE is always acquired. In addition, MANAGER now tests for the presence of 24- or 31-Amode pools, as appropriate for the LOC parameter on the STORAGE macro, and acquires pool storage accordingly (if GETMAIN not forced). Therefore, the RMPOOLS global has also been removed. To prevent problems due to overlaid pool headers, the header is validated for STORAGE (first free block header) and STORFREE requests. If the header of a block being freed is invalid, the block is not returned to the pool's free chain. A snap of the invalid header's pool area is provided, a message is issued, and a thread dump is produced for user debugging (current owner of previous block may have done the overlay), then processing continues. Thus, program checks (S0C4) in MANAGER that eventually bring down the system are avoided.

Core Use Statistics have been expanded to report both 24- and 31-Amode storage/pool requests. If implemented (RMACCT and RMSTATS globals), Request Distribution statistics and Pool Detail statistics are provided for both 24- and 31-Amode pools, if present. RMTRACE has been rewritten to provide these tuning statistics. POOLDUMP has also been optimized to use less print lines/processing time, and modified to print 31-Amode pool headers as for 24-Amode pools. POOLDUMP will recognize an invalid pool header (prints it in Hex if no RCB found for its address).

User programs loaded above the 16M line (executing in 31-Amode) may use the SPA parameter on STORAGE and STORFREE, and the SPAEXT parameter on PASS and CATCH (enter Manager in 31-Amode). If the SPA parameter is used on a STORAGE macro by a 31-Amode program, the LIST parameter may be omitted (and RENT=NO is forced), but if LIST is used then it must point to 24-Amode storage. MANAGER executes in 24-Amode (except when processing 31-Amode pools/storage), but will return in 31-Amode to a caller executing in 31-Amode when the macro was issued. If LOC=ANY is coded on the STORAGE macro, all programs may also use the ADDR parameter to point to a 31-Amode address. Otherwise, omit the ADDR parameter or ensure it is a 24-Amode address (high-order byte is zero), especially if requesting storage with register notation for the ADDR parameter, even if LOC=ANY is also coded.

TRAP has been optimized for header and chain checking and modified to permit the user to define a specific pool size for checking only the free chain and block headers of that area. User code may be easily inserted to test for a known clobber (such as a X'40' (blank) in the first byte of a header) in all headers or blocks, or only in those of a specific size. Also, TRAP (after disabling itself) now goes through standard STAEEXIT processing when it issues its Abend 1369 (snap 122 and thread dump issued, etc.). A TRAP31 version (of the revised TRAP) for the 31-Amode pools has also been created (both modules may not be used in the same link).

INTTABLE has been modified to use the 31-Amode pools, if present and desired, for Table (and Page) Facility processing: ensure one 96-byte pool is defined for table access snap processing (if activated).

Documentation Changes: Basic System Macros
 Operating Reference Manual
 Assembler Language Programmers Guide
 Messages & Codes

New External Options: see above.
 Sample IC31PL00 provided.

Installation: After installing 31-Amode Intercomm pools support, then:
 code an IC31PLnn pools module if desired, then include
 POOLSTRT in the Intercomm linkedit (WARNING: if done in
 a test region with no 24-Amode pools, include in the
 Intercomm linkedit a pools module with at least one
 ICOMPOOL macro for 1 block to prevent POOLSTRT from
 first attempting to load a 24-Amode pools module).
 Ensure IC31PLnn module to be loaded was linked with
 AMODE=31 and RMODE=ANY and that it is linked into one
 of the libraries specified for the STEPLIB concatenation in
 the Intercomm Execution JCL.
 Code SPALIST macro, IC31PL parameter, to specify the
 one- or two-byte alphanumeric name suffix of the IC31PL
 pools module, and reassemble and link the SPA module.

2.3 GENERAL SYSTEM IMPROVEMENTS

Miscellaneous system improvements include:

- STARTUP3 - code to make Intercomm nonswappable incorporated (executed if MRSVC or IISVC specified in SETGLOBE) which is required for all Intercomm regions. This code is now executed before any subtasks are attached/File Handler initialization, to prevent swapouts and thus speed up startup.
- BTVARB macro - conversational time-out (CONV parameter) may be specified in seconds or minutes instead of timer units (see Basic System Macros).
- MANAGER - provide cross-thread STORFREE checking to issue a message if the storage to be freed was not acquired under the same thread number, and to force a program check if neither thread number is 0 (application x freeing storage owned by thread y). See Messages & Codes (message RM033I). This forces the user to optimize STORFREE processing (use PASS/CATCH macros to control storage ownership and/or correct program logic).
- Page Facility - if a PAGE\$TA or \$TC or \$TH or \$TL command is generated internally (has a sending subsystem code in Message Header), no response is queued to the terminal; useful for clean-up under security when user signs off or under VTAM when user logs off. Generate message via appropriate user exit.
- Multiregion Facility changes - message transfer across regions delayed until startup completes (after user exits called); Release 10 can be installed region by region - MRINPUT adjusts BMN number message header field depending on Release number in SPA (see Installation Guide).
- Backout-on-the-fly - restriction removed for THREDLOG DDQ data set in one region only (each QID now has applicable region number); recovery after thread time-out while waiting for access to a VSAM CI implemented (see Section 2.11).
- File Recovery/BOF/Checkpoint processing problems corrected and code optimized, FR log records printing corrected for record-length, macro-number.
- Error Message Reports (Edit/Output Utilities) now use MMN field label instead of MSN (for MSGHMMN) with 8-byte number field (see Messages & Codes - Chapter 7).
- Edit Utility - when editing formatted 3270 screen input, return invalid SBA message with SBA code to input terminal (see Messages & Codes - Chapter 7).
- Output Utility - return formatting error message to output terminal (prevent CRT hang) instead of control terminal (see Messages & Codes - Chapter 7, messages from Report 6).
- FESEND - new 'Do not cancel CONV timeout' option code (character C) in byte 1 of option parameter for FESEND and FESEND calls. This option can be used when an acknowledgement (to input) or multi-stage message is sent, but the program is still processing to produce the complete (final) output message. When used, the user is prevented from entering a new message (RLSE implied for next message). See Programmers Guides and SNA Terminal Support Guide.

- FAR LOCK parameter - to lock (prevent Select) a file at startup (in case off-line processing for file not completed). See Operating Reference Manual.
- INTLOAD - entries added for MMU, Store/Fetch, Dynamic File Allocation, and Table facilities calls, as well as for FECMRLSE and FECMFDBK calls, and all Assembler macro calls. To reduce dynamic linkedit processing at startup, INTLOAD can be linked with Assembler or PL/1 programs which call Intercomm service routines. INTLOAD loads the address of the called routine from the SPA and branches to it (can also be used in conjunction with the Link Pack Facility). See Operating Reference Manual and Assembler or PL/1 Programmers Guide.
- SPALIST macro - EDITRTN parameter default changed to 9 (to correspond to the nine Intercomm-supplied Edit Utility editing subroutines). 0 may be coded if the Edit Utility is not used in a Satellite Region. See Basic System Macros.
- Macro documentation - additional macros documented in the Release 10 edition of Basic System Macros for user Assembler routines include GETSPA, INTTCB, EXTERM, DDNFIND, EXSS, EXTRT, SCONV, VWTO, VWTOR and XASWITCH. Additionally, the new macros GETDATE and QCOUNT are also documented, as are new/revised parameters of existing macros. See the Macros manual Table of Contents for the complete list of macros available to the user.
- RTNLINK overhead replaced with STORFREE (of save area) if needed, and DISPATCH EXIT in dispatched code.
- Broadcast messages passed to the Output Utility with a VMI of X'57' or X'67' are redirected to FESEND for asynchronous processing.
- SYSnnnnn MVS system DD names are bypassed during File Handler initialization.
- Save area chaining: at the end of startup, the MVS save area is down-chained to the Dispatcher's save area which is chained to the save area of the most recently dispatched task (where possible). See Section 2.6.19.
- Log Analysis corrected to allow for 1000's of terminals, 100's of verbs, more than 254 offspring of a subsystem, transaction total over 5 digits.
- RCB display for a core address is eight digits if 31-Mode storage.
- RCB display for an NQ resource shows if exclusively owned (OWNER) or may be shared (SHARE).
- 3390 disk support added (SPINOFF, IXFDYALC, CREATEGF, KEYCREAT, etc.).
- SPINOFF revised to use TRKCALC macro to calculate snap pages (required for 3390 disk support).
- Up to 63 disk queue data sets may be defined per subsystem, BTAM, or VTAM message queue definition group.
- INTTIME macro revised to use issuer's save area to save issuer's registers 2-12 and to use the 3rd thru 7th words of the save area as work areas (except if SAVE=NO is coded on the macro). Therefore, if SAVE=YES (default), the macro can be issued within a reentrant program, which obsoletes the BUSY parameter.

- Dynamic linkedit optimized to build Csect and Entry-point name/address table in 31-Amode core (also used by SCTL command). DYNLWORK data set eliminated. Ensure ICOMDYNL in linkedit to build table at startup via loaded subtask ICOMCESD, even if all subsystems and subroutines are resident.
- Address/name (Csects and Entry-points) table for IJKWHOIT moved to 31-Amode core (also used by SCTL command). See Messages & Codes, Chapter 6.
- BINSRCH can be called for 31-Amode tables (used by several Intercomm modules).
- ICOMFEOF support for 3480 tape cartridges added.
- DDQ SCF file blocksize may be up to 32760 bytes (also increase DDQDS macro FETSIZE parameters to match, and recreate PMIQCFDD file).
- Dispatcher Wait Queue search to add an Event WQE optimized (address of last external ECB Wait WQE saved/updated).
- Multiregion Q flushing logic corrected.
- PMISNAP macro - new parameter XA=YES/NO to indicate some storage addresses in the snap list are 31-bit; macro supported for batch mode processing if PMISNAP1 and BATCHPAK included with batch program, and calling program declares an EXTERN PMISNAP, and opens the PMISNAP DCB for output (requires a SNAPDD DD statement in execution JCL).
- LOGMERGE utility provided to merge Intercomm log data sets when flip/flop log data sets on disk are used (see Section 2.6.15).

2.4 USER EXIT CHANGES

The following lists user exits for which calls have been added, or for which sample exit routines are supplied or processing changed:

- COPYEXIT - sample provided for BTAM/VTAM COPY command processing and AID key support.
- USROTEDT - when called by FESEND, no snap or message issued if exit rejects output message.
- USERB37E - sample provided to unload log data sets when flip/flop facility used (avoid operator responsibility).
- PREPROGI and PREPROGE - new exit calls to add user areas for COBOL Linkage Section and process added areas on subsystem completion.
- USRPAGEX - sample provided to control adding of pages to Page Facility responses and to control creating responses.
- USRQMONX - sample provided to determine receiving Satellite Region (under Multiregion) when input terminal not locked to a region.
- MRSECUR2 - new exit called by MROTPUT in a Satellite Region to determine if message can be passed to Control Region.
- SECUEXIT - sample provided for ESS processing.
- LUCUR - sample provided for VTAM HALT exit to ensure ESS user signed off when session lost/user logs off without signoff.
- USRSTSCH - sample provided to process generic Station Table entries when EXTERM for a terminal-id fails to find a specific entry.

Desupported user exits - the following user exit calls have been deleted:

- TCAMSTRT - formerly called by STARTUP3 for Basic TCAM initialization.
- TCAMCLSE - formerly called by CLOSDWN3 for Basic TCAM completion.

2.4.1 COPYEXIT User Exit

A COPY command processing user exit call is now also provided for VTAM 3270 processing. To facilitate the use of an AID key for general COPY processing, a sample COPYEXIT is provided which is valid for both BTAM and VTAM. A COPYEXIT routine can be used to reject the copy request (non-zero return code), or substitute a different destination terminal than that requested via the input command. To accomplish the latter, the sample exit supports two versions of a generalized COPY command:

- 1) COPY\$\$MYPRT - a table (coded using MYPRT macros) called MYPRTTAB is searched for the entering terminal name. If found, the corresponding 'to' terminal name in the table is substituted for the MYPRT psuedo-name. See the sample MYPRTTAB provided on SYMREL.
- 2) COPY\$\$DUMMY - the input terminal Intercomm name is substituted for DUMMY and the last character is changed to a P and assumes a terminal of that name exists in the Front End (input 'from' terminal name TERM1 becomes output 'to' terminal TERMP).

Documentation Changes: SNA Terminal Support Guide
BTAM Terminal Support Guide

New External Options: MYPRTTAB table coded using MYPRT macros.

Installation: AIDDATA macro coding for general COPY command.
Include sample or user-coded COPYEXIT in Intercomm
linkedit. Include MYPRTTAB table in linkedit, if coded.

2.4.2 USROTEDT - FESEND Call

FESEND has been modified so that if the user-coded USROTEDT user exit rejects (non-zero return code) the output message, no snap 53 or error message (MG600I) is issued. Neither is the output message logged. Note that for VTAM, the OTQUEUE user exit has the same function, but F2 and F3 log records are created.

Documentation Changes: Operating Reference Manual

New External Options: None.

Installation: Include USROTEDT with FESEND, if coded.

2.4.3 USERB37E User Exit

This exit routine is called by IXFB37 which implements the BSAM file flip/flop facility. The intent of the exit is to bypass operator responsibility to submit a job to unload the 'flipped-from' data set before it needs to be reused. Also, coding the exit can be used to avoid issuing a WTOR requesting an operator reply when the unload job is completed. The provided user exit attaches a subtask to asynchronously perform the unload (Intercomm log data sets only) and then post the ECB, which the WTOR would have waited on, to indicate unload completion so the data set can be reused. This processing allows for fairly small log data sets (INTERLOG and INTERLOC) and thus minimizes recovery after a system crash. Closedown waits for the unload to complete (ECB posted), if necessary.

Documentation Changes: Operating Reference Manual

New External Options: None.

Installation: Include sample exit routine with IXFB37, or modify to site needs, if desired.

2.4.4 PREPROGI and PREPROGE Exits

PREPROGI is called by PREPROG to allow the site to add parameter areas to the COBOL subsystem Linkage Section definitions (after the standard Intercomm area addresses of the input message, SPA, SCT, return code, DWS). Up to five area addresses can be added to the parameter list whose address is passed via register 1. On normal return to PREPROG from the subsystem, the exit PREPROGE is called, passing the address of the modified parameter list. The second exit can be used to clean up after the subsystem (free any areas acquired for the thread). PREPROGE is also called by RMPURGE (before resource cleanup) if the subsystem times out (after 'hung' thread recovery, if needed) or program checks.

Documentation Changes: Operating Reference Manual

New External Options: None.

Installation: Code and include user exits with PREPROG, if desired.
Applies only to 'reentrant' subsystems (LANG=RCOB), including VS COBOL II.

2.4.5 USRPAGEX User Exit

This new exit is called by the Page Facility routine PAGE to validate whether the new message passed to PAGE can be added to the current Response Table or if a new response can be created for the terminal in the message header. This exit can thus control the number of pages within a response and can also verify whether the requesting user is still signed on at the input terminal. The parameters passed are the address of the message to be added and the address of the Page Master Table entry for the terminal (see PGEDSECT copy member for Dsect).

A sample exit routine is provided which checks whether the MMN number for the new message is lower than that for the current response. If so, the message is rejected (terminal user started a new response before current was completed). The exit also checks, if ESS is in use, if there is still a user signed on at the input terminal. For Basic Security, the same test is made, however this test is only valid for a single region Intercomm system. If the exit rejects the new message, the return code must be 20, otherwise 0. See comments in sample exit routine.

Documentation Changes: Page Facility

New External Options: None.

Installation: Include sample USRPAGEX with PAGE, or modify, as desired.

2.4.6 USRQMONX User Exit

Under a Multiregion environment, this user exit is called by MRQMNGR if RAP is not used, or the terminal is not currently locked to a Satellite Region and the receiving subsystem for an input message is defined in more than one Satellite Region via the RDT. In this case, it is random from one execution to the next as to which region the message will be passed to. The addresses of the input message, the SUBSYS entry arrived at (after binary search of subsystem codes) and associated REGION entry are passed. The exit can find a different SUBSYS entry for a different region and pass back those addresses, reject the message, or permit queuing to found region.

A sample exit routine is provided which always queues the message for the first region for which the subsystem code is defined. See comments in USRQMONX.

Documentation Changes: Multiregion Support Facility

New External Options: None.

Installation: Include provided exit with MRQMNGR in Control Region, or modify, as desired.

2.4.7 MRSECUR2 User Exit

This exit is called by MROTPUT in a Satellite Region and can be used to determine if an output message should be passed to the Control Region, thrown away, or modified before being passed. The exit must free the message (address passed in R1) and return a non-zero return code in R15, if it is not to be passed. Otherwise, return the address of the revised or original message in R1 and a return code of 0 in R15.

Documentation Changes: Multiregion Support Facility

New External Options: – None.

Installation: Code user exit if desired, and include with MROTPUT subsystem in applicable Satellite Region(s).

2.4.8 SECUEXIT and LUCUR ESS Exits

These exits are documented in detail in the ESS manual. Samples are provided.

Documentation Changes: Extended Security System

New External Options: None.

Installation: See ESS manual.

2.4.9 USRSTSCH User Exit

If Basic Security or ESS is not implemented, generic entries for input/output (CRT) terminals in the Station Table (PMISTATB) may be used in conjunction with this exit which is called by PMIEXTRM if an EXTERM fails to find a Station Table entry for the specified terminal name. Even if security is installed, generic names can be used for all, or groups of, printers. A supplied version of this exit processes for both CRT's and printers, and can be modified for installation naming conventions.

Documentation Changes: SNA Terminal Support Guide

New External Options: Modify supplied USRSTSCH for site, if desired.

Installation: Include USRSTSCH in Intercomm linkedit.

2.5 DESUPPORTED FACILITIES

The following are desupported because they are no longer applicable or are obsolete (see also list of deleted modules in Appendix A):

- OS/MFT and OS/MVT operating system support
- VS1 operating system support (VSSYSTEM global removed)
- SVS and VS2 operating system support (VS2 global removed)
- System/360 operation code restrictions (SYS370 global removed)
- MVS/370 (SP 1) operating system support (MVS and XA globals removed)
- Pre-ESA Operating System tests: all SPA/SPAEXT flags set on at startup for MVT, VS, MVS, 370 and XA, then ESA is tested for
- Page Fixing - not supportable/needed for XA/ESA
- Basic TCAM support (TCAM global removed - only Extended TCAM via GFE supported)
- DISAM, AMIGOS and CFMS Access Methods (AMIGOS global removed)
- OS RJE Facility (code removed)
- Startup WTO parm and SYCTTBL WTO parameter (not needed)
- Page Facility PAGETBLE and BDAM Page data sets (not used)
- CICS macro-level program support
- SPIE/STAE processing (replaced by ESPIE/ESTAE)
- STATION macro labels/Company numbers (ignored)
- DYNLWORK data set (no longer used for dynamic linkedit)
- Obsolete SPALIST parameters: FIX, MBPR, MSPR, NWAIT, TITECOR, SECRTY, SIGN, SNAP, TITV, TVTH, TPMOD, WTO; also CO=NO (default), DVASN=NO (default), and DUPL=NO (default) may not be changed
- PL/1-F and Assembler F support
- SYCTTBL macro - BLRI and DCBP parameters, SPAC parameter applies only to PL/1 subsystems (define ISA size) - and is ignored for other subsystem types
- TUNERTBL for Fine Tuner commands (not used)
- IGCFAST module and FASTSVC global (removed)
- KEYFLIP module and Dispatcher EXECWAIT Csect

2.6 PROGRAM PROCESSING OPTIMIZATIONS

The following enhancements represent processing optimization, and ease maintenance and installation, for user subsystems and subroutines:

- GENESIS3 subsystem initialization incorporated in STARTUP3
- OVLY/EXGRP subsystem processing optimized
- SYCTTBL disk queuing overhead reduced
- Storage availability checking eliminated
- SYCTTBL macro processing parms added
- Indicative dumps controllable by ID and subsystem
- DYNREQ1 sequential search eliminated (COBOL subsystems), maximum DWS size for a COBOL subsystem increased
- Dynamically loaded subsystem loading overhead reduced
- Dynamically loaded subroutine loading overhead reduced
- Loading above 16M line supported for COBOL programs
- VS COBOL II supported
- FINTUNER commands rewritten and enhanced
- 'Program Cancelled' message changed
- PMISERC3 optimization
- PMIDCB eliminated - DCBs moved to processing modules
- FASTSNAP processing changed
- Automated message restart facility
- 'Hung' thread recovery to free resources after eventual I/O completion
- After 909 (IJKTLOOP) abend, thread resources not purged until outstanding I/O completed, if any
- Prevent new message start of non-reentrant (MNCL=1) subsystem while previous thread 'hung'
- SAM (System Accounting and Measurement) also supports statistics on CALLs to COBREENT, PAGE, and the new Table Facility entry-points

- Address of the parameter list passed to a subsystem (from PREPROG if COBOL) has been added to the INTTCB area
- Save area down-chaining from the MVS save area implemented to enhance debugging
- Loading above the 16M line supported for Assembler Language programs
- Loading above the 16M line supported for PL/1 programs
- Direct calls via INTLOAD from loaded Assembler or PL/1 programs to user subroutines defined in REENTSBS supported
- Subsystems loaded above the 16M line process as though resident (continuous multithreading of up to MNCL messages)

2.6.1 Subsystem Initialization and Control

These enhancements consist of the following:

- 1) Code formerly in GENESIS3 to dispatch the Subsystem Controller for each resident/loadable subsystem, and for each subsystem defined in the first OVLY/EXGRP has been incorporated into STARTUP3.
- 2) For subsystems defined with a non-zero OVLY or EXGRP parameter, the ascending numeric sequence checking is now done during assembly of the SCT, instead of by CKOVLYNO during Intercomm startup.
- 3) Code formerly in TRAFFICQ to determine OVLY/EXGRP subsystem group changing and dispatch of Subsystem Controller for the group has been optimized and moved into the Subsystem Controller (SYCT400). Call from Message Collection (BLMSGCOL) deleted. BLMSGCOL will cause an OVLY/EXGRP change when necessitated by input message queuing.

The above changes eliminate the possibility of startup processing problems, extra unnecessary calls and storage requests.

Documentation Changes: Operating Reference Manual

New External Options: None.

Installation: Includes for GENESIS3, CKOVLYNO, and TRAFFICQ deleted from ICOMLINK.

2.6.2 Subsystem Disk Queuing Always FIFO

To reduce disk queuing overhead and flag testing in Message Collection and the Retriever, all disk queuing is now in FIFO order. The BLRI parameter to specify non-FIFO queuing has been desupported. If no core queue is specified (NUMCL=0), then the disk queue blocksize need only be large enough to hold the longest message to be queued. Otherwise, the disk queue blocksize (if DFLN parameter coded) should be the average message length times the value coded, plus one, for the NUMCL parameter (core queue refreshed from disk block when core queue empty). The first message in the block is returned to the caller (Front End or Back End) of the Message Retriever. Core pools should be defined to match disk queue block sizes if extensive disk queuing is expected.

Documentation Changes: Basic System Macros
Operating Reference Manual

New External Options: None.

Installation: Review core pool and disk queue block sizes for both Front End and Back End queues as described above. In the SCT, delete the BLRI parameter if coded on SYCTTBLs (parameter ignored).

2.6.3 Storage Availability Checking Eliminated

Under OS or VS1 executing on the System/360 or earlier System/370 computers, where storage was tight and the region sizes now available under MVS could not be defined, it was necessary to determine storage availability before dispatching a new subsystem thread or loading another subsystem. Originally, this checking consisted of a call to GAMFQES to chain through OS subpool queues to determine contiguous storage availability. Under MVS/370, GAMFQES was changed to FQES which returned a dummy value. Because it is now assumed that adequate region sizes can be defined, and that the user defines (and keeps tuned) Intercomm core pools, the question of storage availability is no longer applicable. Therefore, the calls to FQES (GAMFQES) have been eliminated and the TALY\$SU command display no longer shows core availability (not meaningful under XA or ESA). In addition, the SYCTTBL SPAC parameter is now only used to define ISA storage to be acquired for PL/1 subsystems. Also, the SPALIST MBPR and MSPR parameters are no longer needed (ignored).

Documentation Changes: Basic System Macros

New External Options: None.

Installation: FQES (GAMFQES) no longer in linkedit. Omit SPAC parm on SYCTTBLs except for PL/1 subsystems. Omit MBPR and MSPR from SPALIST macro coding.

2.6.4 Additional Subsystem Processing Parameters

Several parameters have been added to the SYCTTBL macro for subsystem processing control. These are:

- SAM (YES/NO) - if SAM statistics gathering installed, it can be eliminated for specific subsystems by coding SAM=NO, thus statistics for only changed/new subsystems could be gathered, as desired. Selective gathering saves processing overhead.
- REJECT (YES/NO) - reject (do not queue) a new message for the subsystem if processing has been delayed (via DELY command) for the subsystem, and return a 'TRY LATER' message to the terminal. Previously queued (before the DELY) messages remain on the queue, but can be flushed (see FINTUNER below).
- DWSCHK (NO/YES) - for COBOL subsystems, DWS overflow checking is performed only for specific subsystems (provided that 'DWSCHK=YES was coded on the SPALIST macro). Global checking can be controlled via the STRT/STOP commands.

Documentation Changes: Basic System Macros
COBOL Programmers Guide
Operating Reference Manual
System Control Commands

New External Options: SYCTTBL macro: SAM, REJECT, DWSCHK parameters.
Dynamic control: FTUN/SSUP commands
(see Section 2.6.10).

Installation: Code new parms, as desired, on subsystem SYCTTBL macros.

2.6.5 Indicative Dump Control by ID and Subsystem

Previously, producing indicative dumps for 114, 118, and 126 snaps was only a global option. Now, coding of the SPALIST INDUMP parameter allows specification of which of these dump types are to be indicative (thread-related data), and which are to be full region dumps. For example, in a test environment, full region dumps for 126 snaps (program check) may be needed, while only indicative dumps are needed for 114 (enqueue time-out) and 118 (thread time-out) dumps. In addition, even though indicative dumps are specified in the SPA (for some or all IDs), they can be suppressed (force full region snap) for individual subsystems by coding of a new SYCTTBL parameter:

INDUMP (YES/NO) - if YES (default) is coded, then indicative dumps will be produced depending on the SPALIST macro, INDUMP parameter coding.

Global indicative dump control (all or specific IDs) is provided by new options on the STRT/STOP system commands.

Documentation Changes: Basic System Macros
 Operating Reference Manual
 System Control Commands

New External Options: SPALIST macro: INDUMP parameter (revised).
 SYCTTBL macro: INDUMP parameter.
 Dynamic INDUMP control: STRT/STOP, FTUN commands.

Installation: Revise SPALIST INDUMP parameter coding, if desired.
 Code INDUMP=NO on SYCTTBL macros for subsystems
 where full region dumps are desired.
 Note: existing SNAP parameter on SYCTTBL macro can be
 used to suppress 118 (timeout) snaps.

2.6.6 COBOL Subsystem GET/FREE Optimization

For reentrant COBOL subsystems, when the GET (and FREE) parameter was coded on the SYCTTBL, a separate Csect (DYNREQ1) was generated containing entries for each of these subsystems. Each entry contained the subsystem code and the GET and FREE values. To initialize thread processing, it was necessary for PREPROG to sequentially search DYNREQ1 for the corresponding entry to determine the amount of DWS storage to obtain (GET value). The Csect and the overhead of the search have been eliminated by storing the values directly in the SYCTTBL area. This was made possible by revising the SPAC parameter as previously discussed under Storage Availability Checking (see Section 2.6.3).

NOTE: DWS sizes may now be up to 64K less 308 bytes.

Documentation Changes: None

New External Options: None.

Installation: Automatic (reassemble SCT at Release installation).

2.6.7 Dynamic Subsystem Loading Optimization

Dynamically loaded (below the 16M line) subsystems which are serially reusable or reentrant will be kept in core if possible, rather than forcing a delete and reload every time MNCL messages have been processed. The loaded subsystem is kept in core after MNCL messages have been started provided the following conditions are met:

- 1) Either other messages are still being processed, or
- 2) No program problem (program check, time out) or reload request (LOAD command) has occurred, or
- 3) Processing has not been delayed (DELY command), or
- 4) CONVERSE or INITLU6 responses are expected, or
- 5) Storage Cushion not released (no low core condition exists), or
- 6) Additional messages are queued, and
- 7) No other subsystem is waiting to be loaded (below the 16M line) because the SPALIST MAXLOAD value has been reached.

This optimization saves processing overhead for high-volume loadable subsystems.

Documentation Changes: Operating Reference Manual.

New External Options: MAXLOAD value, and number of times value reached, now displayed via TALY\$SU command.

Installation: Automatic.

2.6.8 Dynamic Subroutine Loading Optimization

For dynamically loaded subroutines which are defined on SUBMODS macros as serially reusable or reentrant (USAGE parameter), the DELTIME parameter now allows a value up to 60 minutes for an inactivity time before the subroutine is deleted. Coding a value greater than 60 seconds (old DELTIME maximum value) allows frequently used subroutines to remain in core. (Reloading can be forced via LOAD command.) This extended time reduces the potential for core fragmentation. Forced reload via the LOAD command is also supported for loadable subroutines defined with PERMRES=YES: if dynamic linkedit of the new version fails, the old copy will be kept in core for continued use. If not PERMRES=YES, the old copy is marked unusable when dynamic linkedit fails after a LOAD command is issued.

Documentation Changes: Basic System Macros
Operating Reference Manual.

New External Options: None.

Installation: Revise DELTIME parameter on SUBMODS macros in REENTSBS for applicable subroutines, if desired.

2.6.9 COBOL Program Loading Above the 16M Line

PREPROG, COBREENT, and DYNLOAD have been revised to process dynamically loaded 'reentrant' COBOL subsystems and subroutines which are loaded above the 16M line. These programs must be compiled with the OS/VS COBOL compiler (or equivalent) and linked with the RMODE=ANY and AMODE=31 attributes. Programs to be loaded above the 16M line must strictly follow the Intercomm pseudo-reentrant coding conventions, plus all parameter values (except the REENTSBS code) passed to Intercomm or user routines (via COBREENT) must be coded in the Intercomm-provided 24-Amode DWS area. Therefore, Map names, File DD names, etc. which are hard-coded in the WORKING-STORAGE SECTION must be copied at execution time to fields in the DWS (must be 24-Amode addresses) and the DWS field names must be used for calls to COBREENT (which is entered/returns in 31-Amode).

Related Intercomm modules such as IJKWHOIT, and PMISNAP1 have been modified to recognize 31-Amode addresses, in addition to the program loading and program error recovery modules. Reentrant COBOL subsystems actually loaded above the 16M line remain loaded unless a program error or forced deletion occurs, and can continually process MNCL messages (no slowdown in throughput occurs).

Documentation Changes: COBOL Programmers Guide
Operating Reference Manual
Basic System Macros.

New External Options: SAM tracking of COBREENT calls (see Section 2.6.17).

Installation: Revise COBOL programs as necessary according to COBOL Programmers Guide, relink with RMODE=ANY, AMODE=31. Do not use VS COBOL II compiler until that support has been installed.

2.6.9.1 VS COBOL II Support

COBOL programs compiled with the IBM VS COBOL II compiler (Release 3.0 and up) are supported as long as documented restrictions and requirements are adhered to. The programs must be coded as reentrant, call all service routines and user subroutines via COBREENT, and may be resident or loadable above or below the 16M line. Processing for loading above the 16M line is the same as for OS/VS COBOL programs, except that fixed values passed as parameters on calls to COBREENT may be in the WORKING-STORAGE SECTION. A mixed OS/VS and/or ANS COBOL and VS COBOL II environment is supported except as documented.

Documentation Changes: COBOL Programmers Guide
Installation Guide
Messages & Codes
Basic System Macros
Operating Reference Manual

New External Options: CB2SNAP parameter: SPALIST macro and STRT/STOP commands (to produce interface snaps, if desired).

Installation: See above documentation.

2.6.10 FINTUNER Command Revisions

The Fine Tuner group of programs provides commands for dynamic control of subsystem processing, and dynamic update of modifiable subsystem SYCTTBL parameters. To ease command processing, the previously required TUNERTBL has been eliminated. Instead, the first command parameter is the codes of the subsystem SYCTTBL to be modified, in the form (ssch\$ssc), as previously implemented for the TALY\$BE command. The Fine Tuner commands are:

- BEGN - resume processing by a previously delayed/stopped subsystem
- DELY - stop processing new messages for a specified time
- MNCL - modify MNCL (message thread concurrency) value
- PRTY - modify PRTY (execution priority) value
- SPAC - modify COBOL GET/FREE (DWS size) or PL/1 SPAC (ISA area size) values for new messages
- SSFL - flush 1, n or ALL queued messages
- TCTV - modify TCTV (time-out) value
- FTUN - display current subsystem SYCTTBL values, message counts, etc.
(2 screens generated via MMU)
- SSUP - change 1 or more modifiable values in the second FTUN screen.

The FTUN and SSUP commands are processed by a new subsystem called DYNSSUP, and require that MMU and Store/Fetch modules be included in regions where the commands may be executed (Store/Fetch MMU data sets not required).

Documentation Changes: System Control Commands.

New External Options: New commands listed above (see also Section 2.15).

Installation: Ensure FINTUNER, TUNRBEGN, RPT00009, DYNSSUP included in linkedit (automatic via ICOMLINK).
 Ensure SYCTTBLs for FINTUNER and DYNSSUP defined (supplied in INTSCT, etc.).
 Ensure subsystem codes for FINTUNER (SUBC=T) and DYNSSUP (SUBC=F) added to PMIRDT if Multiregion used.
 Ensure Fine Tuner verbs in BTVRBTB, if supplied version not used (remove from COPY member USRBTVRB if supplied version used).
 Ensure MMU and Store/Fetch modules in linkedits for regions where FTUN may be used.

2.6.11 'Program Cancelled' Message Improvements

A more informative message has been created via the supplied USRCANC (PMICANC) user exit to list the subsystem code (in hex and alpha) and the subsystem name on the first line, followed by the detailed 'reason' message on the second line. (RPT00008 also modified for new message format.)

If a new message is cancelled (CANC=STOP coded/set in SYCTTBL and a previous problem occurred) or flushed (see SSFL command above), an appropriate response message is formatted via RPT00009 containing the input transaction code (first four bytes of message if pre-edited).

Documentation Changes: Operating Reference Manual.

New External Options: None.

Installation: Ensure PMICANC, and RPT00008 and RPT00009 (for Output Utility) are in the Intercomm linkedit (automatic via ICOMLINK).

2.6.12 PMISERC3 Utility Subroutine Optimization

PMISERC3 is called for Utility (Edit, Output, Change/Display) processing. Optimization consists of combining and reducing entry points and related processing, and changing GETMAIN/FREEMAIN code to use the Intercomm STORAGE and STORFREE macros, and to optimize their use.

Documentation Changes: None.

New External Options: None.

Installation: Automatic.

2.6.13 PMIDCB Module Changes

Due to the desupport of the Basic TCAM interface (see BTAM/TCAM Changes in Section 2.8), the coding of system DCBs in a separate module (PMIDCB) is no longer necessary. The three existing DCBs are handled as follows:

- QTAMDCB deleted (Basic TCAM only)
- PMISNAP moved to open code in PMISNAP1 (ENTRY added)
- FASTSNAP moved to SVC dependent code in PMISNAP1 (ENTRY added).

The PMIDCB module is therefore obsolete.

Documentation Changes: Operating Reference Manual.

New External Options: None.

Installation: Ensure PMIDCB not in Intercomm linkedit.

2.6.14 FASTSNAP Processing Changes

To reduce the number of Intercomm SVCs requiring installation, Fastsnap processing has been modified to use an Intercomm Interregion SVC (MRSVC global, IGCICOM module or IISVC global, IGCICSVC module). The FASTSVC global has been deleted, along with the IGCFAST module. Also, the blocksize in the FASTSNAP DCB (see above) is internally corrected if executing under ESA (not XA).

Documentation Changes: Operating Reference Manual
Installation Guide

New External Options: None.

Installation: Ensure IGCICOM or IGCICSVC installed, and that KEYFLIP is deleted from the Intercomm linkedit.
If IGCFAST previously installed, delete it from MVS nucleus.
Do not ORDER KEYFLIP, nor EXECWAIT Csects (both obsolete).
Note: Fastsnap processing also requires a FASTSNAP DD statement in the Execution JCL to be activated.

2.6.15 Automated Message Restart Facility

This facility was provided by a user and described in detail at a recent User Group Meeting. Its design is to preclude operator intervention (other than resubmitting the Intercomm job) in the event of a system failure requiring message restart processing. One small (20-byte record) BDAM file (STRTUPSW) is used to track the status of the last Intercomm execution (successfully closed down or failed). If message restart is implemented, the file is examined at startup by the new module AUTORCVR, to determine whether a restart is needed (EXEC parm ignored). This facility assumes Intercomm logging is to flip/flopped disk data sets, or to a single disk data set large enough to hold all messages for normal (no failure) processing. A new module, LOGMERGE, is provided to unload the Intercomm log data set(s) to the message restart data set RESTRTLG. If LOGMERGE determines from STRTUPSW that message restart is needed, it attaches the Intercomm utility ICOMFEOF to insure valid log files. If message restart is not needed, then only the log merge is executed. Therefore, LOGMERGE must be executed before the Intercomm step, to implement this automated facility. As an adjunct to this facility, if flip/flop log data sets are used, the supplied user exit USERB37E (described above) can be used on-line to unload a flipped log data set to the disk data set used later for RESTRTLG.

Documentation Changes: Operating Reference Manual
 Messages & Codes
 Installation Guide (generate JCL to initialize STRTUPSW data set via new utility AUTORSET, if CHKRES=YES on ICOMGEN macro).

New External Options: None.

Installation: Set up JCL to execute LOGMERGE.
 Create the STRTUPSW data set.
 Include AUTORCVR with message restart modules (automatic via ICOMLINK if CHKRES=YES).
 Code PARM=STARTUP for Intercomm EXEC statement.

2.6.16 'Hung' Thread Recovery

If a message processing thread times out (value coded for TCTV on subsystem's SYCTTBL elapses) while the thread is disabled, then RMPURGE causes an internal wait to occur to allow the thread to become enabled for purge. The internal wait time is again TCTV time. A thread is disabled/enabled by system routines such as the File Handler to allow I/O to complete (prevent purge of data record area/control blocks), or may be disabled in a Data Base interface to allow data base I/O to complete, or by Message Collection to allow disk queuing of a message to complete, etc. If the second wait interval expires without the thread being enabled for purge, then the thread goes into HUNG status, and only NQ(WAIT) RCBs are purged. Formerly, all outstanding WQEs (Dispatcher Work Queue Elements) for the thread were also purged, which would never allow the thread to be enabled (free owned resources). Now, only timer and internal wait WQEs are purged: external wait WQEs (whether on the Wait queue or an Execute queue) are not purged so that the routine that issued the disable can finish and enable the thread for purge at a future time. When the disable count is zero, RMPURGE is given control to finally free all the thread's owned resources (including NQs, storage, etc.).

If a subsystem's original TCTV time is generous enough to allow for the thread's normal I/O processing, then a HUNG situation should not occur. However, Cross-system RESERVEs or disk channel access path or tape mount problems can cause a delay in completing I/O, which this recovery processing is designed to handle. It also provides recovery (prevents storage overlay) for:

- 1) Single-thread subsystem processing (subsystem not reentrant or MNCL is 1), and
- 2) IJKTLOOP timeouts (recovered 909 Abends) when the thread was disabled and not in a code loop. The thread may be looping in access method OPEN processing or when non-VSAM PUT processing is used for a sequential file.

In case 1, a new thread for the subsystem is not started until the old thread is completely purged; if the subsystem is in an Overlay link (not recommended), then Overlays are not eligible for change until purge completes. In case 2, processing is the same as for other I/O delays as described above, even though the thread is not in an ECB/CI/DB wait state.

System messages are issued when a subsystem thread becomes 'hung' (RM016I), and when it finally becomes enabled for purge (RM034I). The terminal operator is informed that the input message was cancelled due to timeout (via PMICANC, if in the linkedit - see Section 2.6.11) when the original timeout occurs. A 'hung' thread does not affect MNCL message processing for reentrant subsystems (but may cause other thread timeouts if it holds an Exclusive Control NQ which has not independently timed out).

Documentation Changes: Operating Reference Manual
Messages & Codes

New External Options: None.

Installation: Automatic.

2.6.17 SAM Enhancements

The System Accounting and Measurement (SAM) feature of Intercomm for statistics on subsystem activity (CPU time, storage usage, service routine calls, etc.) has been enhanced to add the following activity tracking:

- Calls to COBREENT by COBOL subsystems
- Calls to PAGE (directly by a subsystem, or via MMU MAPEND option P)
- Calls to Table Facility entry-points.

To have these activities tracked for all subsystems (for which SAM=YES (default) is coded on their SYCTTBL macros - see Section 2.6.4), add the following keywords to the MAPACCT macro definition in the SAMTABLE, as desired: COBRENTS (for COBREENT calls); PAGECLS (for PAGE calls); TABxxxxx (for Table Facility calls).

Documentation Changes: Basic System Macros
 Operating Reference Manual
 Page Facility
 Table Facility

New External Options: See above.

Installation: See ORM for SAM installation.

2.6.18 Subsystem Parameter List Accessibility

For user debugging and exit (PREPROGE - see Section 2.4.4) processing, the address of the parameter list passed to a subsystem thread is stored in the Intercomm Thread Control Block (ITCB) in the new field ITCBPMSS. For Assembler and PL/1 subsystems, this is the address of the original parameter list (as passed to PREPLI if PL/1), and therefore the address is cleared after a thread becomes 'hung' (see Section 2.6.16), and is no longer available. For all COBOL subsystems, this is the address of a parm list area in PREPROG's save area, therefore the fifth word in the list is the address of the DWS area passed to the subsystem. Because PREPROG's save area is not freed until a COBOL thread can be completely purged, this parameter list is still available to PREPROGE for thread cleanup after a 'hung' subsystem is enabled for purge.

Instead of backchaining through save areas, user Assembler programs (interface routines) can easily acquire the address of the ITCB via the INTTCB macro, and then reference (load parameter list address from) the ITCBPMSS field after establishing ITCB Dsect addressability. COBOL program interfaces must use the INTTCB macro, as the back chain (after a call to COBREENT) is different if the subsystem is VS COBOL II.

With this enhancement, the true parameter list (for COBOL) is snapped in an indicative dump, instead of the original 4-word parameter list passed to PREPROG.

Documentation Changes: Operating Reference Manual
 Basic System Macros (INTTCB macro)
 Messages & Codes, Chapter 6

New External Options: INTTCB macro (ITCBPMSS field in ITCB).

Installation: Revise user Assembler subroutines, interface routines, and/or exit routines, to use the INTTCB macro to acquire a subsystem's input parameter list address.

2.6.19 Save Area Chaining

In all dumps (indicative or full region), to make the MVS 'SAVE AREA TRACE' control blocks meaningful, the MVS Operating System save area is now down-chained to Intercomm save areas. The MVS save area (SA) is pointed to by TCBFSA and is always the first save area in the chain (usually at a low core address below the beginning of the Intercomm load module). At startup, Intercomm places the address of the SPA in the first word of this save area. During startup, the MVS SA is chained to that acquired by the STUOVLY processing Csect which is chained to that of the most recently called initialization routine. At the end of startup, STUOVLY unchains its SA, and rechains the MVS SA to that in the Intercomm Dispatcher. The Dispatcher's SA is easily recognized because saved registers 2-12 always contain the value FFFFFFFD. The entry-point listed above a save area is usually the owner of the next save area down the chain.

When the Dispatcher dispatches an Intercomm 'task' from an Execute Queue, it stores the parameter address passed to that task as the down-chain from its save area. The parameter address is often (but not always) the address of the SA of the dispatched routine (for example, this is always true when an INTWAIT macro is issued). If the parameter address is not that of the dispatched routine's SA, then that routine will usually chain its SA (after it is acquired) to the Dispatcher's on entry.

For program checks causing a snap with ID=126, SPIESNAP has been changed for application threads to chain the address of the Subsystem Controller (SYCTRL Csect, SCNMAIN entry-point) SA for the thread to the Dispatcher SA before creating the snap. SYCTRL's SA is chained at message thread initialization to that of the subsystem (if Assembler using the LINKAGE macro), or of the interface routine (PREPROG if COBOL, or PREPLI if PL/1). For subsystem timeouts (snap ID=118), the SYCTRL (PURGE entry-point) SA is automatically chained to the Dispatcher's. For recoverable abends (IJKTLOOP - 909, VS COBOL II - 10nn), STAEEXIT chains the Dispatcher's SA in the same way as SPIESNAP (if an application thread was executing at time of the abend) before creating the associated snap (ID=121 or ID=123, respectively).

Thus, debugging program problems is much easier with a true SAVE AREA TRACE. Also, saved register 1 in each save area contains the address of the parameter list passed to the next (called) routine, if any, and can easily be found without chaining through save areas in a dump. Unfortunately, for PL/1 programs, save areas in the ISA are not down-chained, only back-chained (see PROCEEDING BACK VIA REG 13 save areas for most recent call and use the higher SA, or use register 13 in the SPIE SAVE AREA or STAE WORK AREA, to back chain through the dump to the first PL/1 save area).

Documentation Changes: Messages & Codes, Chapter 6

New External Options: None.

Installation: Automatic.

2.6.20 Assembler Program Loading Above the 16M Line

SYCT400 (Subsystem Controller) and DYNLOAD have been modified to recognize loading above the 16M line for an Assembler Language subsystem or subroutine, and to switch to 31-Amode before branching to the program and back to 24-Amode on return. To provide an interface to 24-Amode Intercomm service routines and to user subroutines, the AMODE=31, RMODE=ANY Assembler program must be linked with INTLOAD which provides macro and CALL entry-points. INTLOAD transfers control to a new Intercomm-resident 24-Amode routine called SWMODE, which performs mode-switching and save area chaining for the calling program.

To be eligible for loading above the 16M line, the program must be coded as reentrant and acquire a 24-Amode save/work area via the LINKAGE (subsystem) or SUBLINK (subroutine) macros and must free the save area before exit via the RTNLINK macro. All parameters passed to Intercomm routines must have 24-Amode addresses (hard-coded file names, map names, etc. must be copied to the 24-Amode work area for passing as parameters). User subroutines in the Intercomm load module must each be defined with the LNAME parameter on a SUBMODS macro in the REENTSBS table and must be accessed via the MODCNTRL macro, or a special INTLOAD CALL (see Section 2.6.22). Intercomm may not be entered in 31-Amode from loaded Assembler programs except via INTLOAD and SWMODE. The SPA, SPAEXT or LINK parameter may not be used on Intercomm macros, and a called service/user routine address may not be preloaded from the SPA, SPAEXT nor USERSPA for the CALL macro. The requirement to use the INTLOAD/SWMODE interface from loaded 31-Amode programs also permits the true owner address to be stored in an RCB (listed on a thread dump).

Note that Assembler programs executing in RMODE=24, but AMODE=31 (via the XASWITCH macro), must use the XASWITCH macro to return to 24-Amode before issuing any Intercomm macros, or calling any other program. Also, they may not pass a 31-bit address as a parameter to any Intercomm entry-point except for the STORAGE, STORFREE, PASS and CATCH macros (see Sections 2.2.9 and 2.2.12) when the system is at SM Level 2300 or higher.

Documentation Changes: [Assembler Language Programmers Guide](#)
[Basic System Macros](#)
[Operating Reference Manual](#)

New External Options: Revised loaded program INTLOAD interface.

Installation: SWMODE automatically included via ICOMLINK.
Revise macro calls to omit SPA, SPAEXT, LINK parameters, as needed.
Revise CALL statements and passed parameter addresses as described above.
Link loadable Assembler program with INTLOAD and link load module as AMODE=31,RMODE=ANY.

2.6.21 PL/1 Program Loading Above the 16M Line

PREPLI and DYNLOAD have been modified to recognize PL/1 programs loaded above the 16M line, and to perform mode switching before branching to, and on return from, the PL/1 subsystem or subroutine, respectively. To provide an interface to 24-Amode Intercomm service routines for loaded PL/1 programs using the copy member PLIENTRY and direct calls, the PL/1 program must be linked with INTLOAD. As for Assembler programs (see above), INTLOAD transfers control to the Intercomm-resident SWMODE routine for save area chaining and mode switching. Static file names, map names, etc. to be passed as parameters must be copied to variables declared in the DSA area of the program originally acquired for it by PREPLI (must be 24-bit addresses).

Optionally, a PL/1 program may use calls to PMIPL1 to interface to Intercomm service routines and user subroutines (which must be defined in the REENTSBS table). If the PL/1 program uses the PENTRY copy member, and only calls PMIPL1, it does not need to be linked with INTLOAD. PMIPL1 recognizes a 31-Amode caller and performs mode-switching on entry from, and exit to, that caller. All parameters passed to called routines via PMIPL1 must be declared in (moved to) the 24-Amode DSA storage area.

Because Intercomm may not be entered in 31-Amode (except in SWMODE, PMIPL1, and IBM..... PL/1 language-dependent subroutines in the linkedit), direct calls to user PL/1 or Assembler subroutines in the Intercomm load module from loaded 31-Amode PL/1 programs are not permitted. The user subroutines must be defined in the REENTSBS table via SUBMODS macros with the LNAME parameter, and accessed via one of three methods:

- 1) called via PMIPL1 (add name and offset to PENTRY), or
- 2) called via an Assembler interface routine linked with the calling program (see sample BINTFAC in PL/1 Programmers Guide) which issues a MODCNTRL macro for the passed program name to access it via INTLOAD, or
- 3) a direct call to the program name entry in INTLOAD (add name to PLIENTRY list), as described in Section 2.6.22.

Any one of these methods also permits the called subroutine to be dynamically loadable above or below the 16M line, as desired, instead of being resident, and greatly reduces dynamic linkedit time for the calling program.

Documentation Changes: PL/1 Programmers Guide
Operating Reference Manual

New External Options: BINTFAC sample user subroutine interface.
Revised loaded program INTLOAD interface.

Installation: SWMODE automatically included via ICOMLINK.
Define user subroutines in copy member PENTRY or PLIENTRY, as appropriate, and in REENTSBS.
During program initialization, move STATIC literals to DSA variables for parameters to be passed to 24-Amode routines.
If PMIPL1 not used for all calls, link loadable program with INTLOAD and link as AMODE=31, RMODE=ANY: optionally code and also link BINTFAC routine.

2.6.22 Direct Calls to User Subroutine Psuedo-Entries in INTLOAD

To ease restrictions on converting Assembler and PL/1 programs for dynamic loading (above the 16M line), INTLOAD has been enhanced to permit direct calls to user subroutine names defined in INTLOAD. User subroutines, whether resident in the Intercomm load module, or dynamically loadable above or below the 16M line, must first all be defined to the REENTSBS table in the copy member USRSUBS (copied into the released REENTSBS at assembly time).

User subroutines are defined in USRSUBS via SUBMODS macros with the LNAME parameter (and other parameters as needed - see Basic System Macros). The order in which the subroutines are defined is irrelevant if neither PMIPL1 for PL/1 interface, nor COBREENT for COBOL interface, is needed. If either of those routines are called by any high-level language programs, then the order is dependent on the subroutine name definitions in the program copy members PENTRY and ICOMSBS, which contain corresponding offsets into the combined REENTSBS table, as described in the respective Programmers Guides. After the user subroutines are defined in USRSUBS, then reassemble INTLOAD to also copy the same USRSUBS, and link the calling program with the new INTLOAD. Internal SUBMODS macro generation is different for INTLOAD, where the called entry-point contains an ENTRY statement and code to convert the CALL into a MODCNTRL macro expansion for entry into standard INTLOAD interface code. The parameter list (address) passed to the called psuedo-entry point is passed on via the internal MODCNTRL macro code. INTLOAD then passes control to DYNLLOAD (via SWMODE if the caller is loaded above the 16M line), where control is then given to the resident, or loaded, user subroutine. When the subroutine exits back to DYNLLOAD, control is transferred directly back to the INTLOAD ENTRY caller (or via SWMODE if mode switching is necessary).

With this change to INTLOAD's CALL support, it is necessary to reassemble INTLOAD to generate the new entries for new calls when new entries are added to the REENTSBS table. However, it is not necessary to relink unchanged loadable user programs with the new INTLOAD. It is recommended that only 1 production and 1 test version of INTLOAD and REENTSBS be maintained across all Intercomm production and test regions. Note that loadable user Assembler and/or PL/1 subroutines must also be linked with INTLOAD for direct call processing. INTLOAD may not be in the Intercomm load module. Resident Assembler programs use standard calls for resident subroutines and must use the MODCNTRL macro to access loadable subroutines, as previously documented. Resident PL/1 programs use standard calls to resident subroutines and can use the BINTFAC interface program (see Section 2.6.21 above) to access dynamically loaded user subroutines. BINTFAC may be in the Intercomm load module for that purpose. Again, since (the sample) BINTFAC generates a MODCNTRL macro to access the called subroutine, the order of entries in USRSUBS does not matter to it either.

As before, all COBOL programs must use the COBREENT interface. User COBOL subroutines (especially if compiled under VS COBOL II) may not be called by non-COBOL programs.

Documentation Changes: PL/1 Programmers Guide
Assembler Language Programmers Guide
Basic System Macros

New External Options: Reassembled INTLOAD to generate CALL entries.

Installation: See above.

2.6.23 Continuous Multithreading of Subsystems Loaded Above the 16M Line

In contrast to region size limitations on the total area within the Intercomm address space that can be occupied by dynamically loaded subsystems below the 16M line, there is virtually no limit on the size of that area for subsystems loaded above the 16M line. Therefore, it is not necessary to define a SPALIST MAXLOAD value for above the line, nor is it necessary to monitor message activity in order to delete a subsystem that has processed MNCL messages when the MAXLOAD value is reached below the 16M line. This forced deletion is done in order to allow a subsystem waiting to be loaded below the 16M line to actually be loaded and given control, in order not to compromise response time. The deleted subsystem is placed on the load queue for reloading (when space permits) if it still has messages queued to be processed. Otherwise, it is not reloaded (space permitting) until a new input message is queued for it. If the MAXLOAD value is large enough to handle keeping most loadable subsystems in core, then there is little noticeable effect on response time (see Section 2.6.7). Note that even if a subsystem is placed on the delete queue, it is not actually deleted until its space is needed and it is not currently processing messages.

For subsystems loaded above the 16M line, since there is no need to free the space they occupy, then they can continuously process up to MNCL messages (if any), and therefore are treated as though they are resident in the Intercomm load module. Except for the initial load time, there is no difference in response time to their being resident (nanoseconds for mode switching do not count). A subsystem loaded above the 16M line will only be deleted if a program check or timeout occurs in one of its processing threads (in case the load module is compromised), or if a LOAD command has been entered to force a delete and reload.

Therefore, converting subsystems (and subroutines - see Section 2.6.8 on reloading criteria) to be loadable above the 16M line has much to gain (allows reloading of a new/corrected version, does not occupy space below the 16M line), and nothing to lose. Actually, response time may improve, and the chances of storage overlay of the code decreases.

Note that specifying a REGION parameter on the Intercomm EXEC statement applies generally to the size of the region below the 16M line, and that (by IBM default) 32M is made available above the 16M line. Intercomm uses storage above the 16M line for the VSAM LSR pool (if requested), some internally constructed tables (see Section 2.3), the Table Facility (if used), the optimized Page Facility (if used), to satisfy user requests for 31-Amode storage, and for the 31-Amode pools (if implemented). The amount of storage (current and maximum) occupied by loaded programs above the 16M line is given in the System Tuning Statistics (at SM Level 2300). Use the maximum value (as programs are converted for loading above the 16M line) along with other tuning statistics, to calculate the total 31-Amode storage used. Monitor these statistics as the 32M value approaches to determine if your MVS systems programmers must allow for a greater region size above the 16M line for Intercomm execution (via IBM user exits).

Documentation Changes: Operating Reference Manual

New External Options: None.

Installation: Automatic (see other Sections on storage used above the 16M line).

2.7 STATISTICS GATHERING AND SYSTEM DISPLAY UPGRADES

The following enhancements and changes have been made for system statistics printout and display modules:

- New SCTL system control, display, and debugging aid command (described in this section).
- COREACCT macro removed from MANAGER module (used to define core block sizes for storage requests distribution statistics in Core Use Statistics); now to be coded with ICOMPOOL macros in Core Pools Table. See Basic System Macros.
- Core Use Statistics counter overflow recovery: when a counter overflows, the printed field is filled with 9s, and statistics for other fields continue to be gathered/printed. Statistics also added for 31-Amode storage requests.
- Pool Use Detail Statistics: the REQUESTS, FILLED, and ALLOCATED printed column sizes are increased to 10 digits, and the FAILED to 8 digits.
- Thread Dump: subsystem processing information added - status (ACTIVE, HUNG, etc.), disable count, message MMN number, input terminal-id.
- Thread Dump: the Csect name + displacement of the issuer of a SUBLINK macro is listed as the core owner, instead of PMISUBL2.
- Pool Dump: - instead of the hex address, the Csect name + displacement of the pool owner can be printed (set POOLNM global in SETGLOBE to 0 if name not desired - address printed as in Release 9).
- System Tuning Statistics - many lines of information added such as: the current BMN and MMN numbers, region-id (or job name if not a Multiregion system), total messages processed/cancelled(flushed)/queued, counts of Multiregion messages sent and received, total space in use by subsystems loaded below the 16M line/max allowed (SPA MAXLOAD value) and number of times MAXLOAD reached, current and max space used by programs loaded above the 16M line, count of operating system WAITs issued by Dispatcher, minimum WQEs on the Dispatcher's Free Queue, new Page Facility and Table Facility tuning statistics, current SPA STOCORE value and a break down of Store/Fetch processing by data set.
- New closedown statistics (on STSLOG data set) produced by new modules SSRPT (subsystem processing statistics for each subsystem) and SUBRPT (REENTSBS user subroutine processing statistics). Ensure modules included in linkedit (automatic via ICOMLINK).
- DDQ Print Utility - to print in hex and EBCDIC all DDQ records for a specific DDQ data set, or only a specific QID. Optionally print only the QIDs for all DDQ data sets or a specific data set. See Dynamic Data Queuing.

- DWSSNAP service routine - provides for snapping, printing, or displaying selected DWS fields or the entire COBOL program DWS area in hex and EBCDIC. See COBOL Programmers Guide. DWSSNAP is also supported for PL/1 ISA areas. See PL/1 Programmers Guide.

In addition, the following enhancements were made which are described elsewhere in this manual:

- New WHOI and WHOU commands to display terminal AID key data
- New TALY command suboptions for condensed displays
- TALY\$SU display provides system tuning data
- New FILE command STAT option for data set status, processing options
- SMLOG and SYSPRINT may be dynamically deallocated (automatically reallocated) for off-line printing
- FILE\$LSR and File Handler stats printout support new (MVS/XA 2.2, DFP V2.3 and up) CI/buffer pool sizes
- SAM (System Accounting and Measurement) statistics can be requested or suppressed per subsystem.

2.7.1 New System Control/Display Command - SCTL

The SCTL command has been developed to provide system display and debugging facilities, and some system modifications, as follows:

- Display Intercomm core, modules, or system table areas (hex only, or hex and EBCDIC) using a Table name, Csect or Entry-point or Load Module name, or an address
- Convert numbers from hex to decimal and vice versa
- Change certain SPA/SPAEXT field values (STOCORE, MMNCL, MDELY, PFMAXS, and SNAPPGS)
- Produce a thread dump (for a specific or all threads)
- Produce a WQE trace (for 1 or all Dispatcher queues)
- Locate a module's Csect or Entry-Point address if name supplied, or Csect/EP name if address supplied.

Optionally, the output can be routed to another terminal, or a printer (depending on size of output). A complete thread dump or WQE trace is written to the SMLOG or SYSPRINT data sets, respectively.

Documentation Changes: System Control Commands

New External Options: SCTL command.

Installation: Automatic (if supplied BTVRBTB and SCT used).
 SYSCNTL processing subsystem automatically included in
 link with Fine Tuner programs (if ICOMLINK used).
 Ensure subsystem code (SUBC=C is required) for
 SYSCNTL is added to PMIRDT for a Multiregion system.
 Ensure ICOMDYNL is in the linkedit, and that ICOMCESD is
 available on STEPLIB for building the name/address table
 for locate and core display processing.

2.8 BTAM/TCAM FRONT END CHANGES

The following changes and enhancements have been made for BTAM/TCAM terminal support:

- LINEGRP macro - internal change via TIB 45 incorporated.
- BDEVICE macro, new NORLSEM parameter (YES/NO) - for 3270 CRTs: suppress output text response to RLSE command, just send keyboard reset command (WCC).
- BTAMSCTS - can now be internally generated (as for VTAMSCTS) via new BTERM parameters: NUMCL, DFLN, PCEN. If they are coded, the QNUM parameter must be omitted. All BTERMs must have either QNUM or queue definition parms; intermixing not allowed. Internal generation forces dedicated queues for all terminals and therefore, queue checking at startup is bypassed in BTAMLINE (reduces startup overhead). Code PCENSCT after PMISTOP to generate disk allocation table data.
- BLINE macro, new WAIT parameter (YES/NO) - wait for startup to complete before dispatching line handler for terminal I/O.
- BTVVERIFY - interval timer test deleted (obsolete).
- FLSH command - flush 1, n, or ALL messages (dedicated queues only), and chaser messages eliminated - supported for leased line terminals and CPU console only.
- Desupported facilities/terminals - Graphics (GAM) Access Method; Basic TCAM interface (TCAM global deleted from INT/SETGLOBE); Wiltek, Bunker Ramo, Uniscope, Sanders terminals.
- Note that the Extended TCAM interface via GFE is still supported.

Documentation Changes: [BTAM Terminal Support Guide](#)
[TCAM Support Users Guide](#)
[Basic System Macros](#)
[Messages & Codes](#)
[System Control Commands](#)

New External Options: BDEVICE macro: NORLSEM parameter.
 BTERM macro: NUMCL, DFLN, PCEN parameters.
 BLINE macro: WAIT parameter.
 FLSH command: n messages.

Installation: Automatic.

2.9 VTAM FRONT END CHANGES

The following changes and enhancements have been made for VTAM support:

- Change decks to support the LU6.2 External Feature are incorporated in existing VTAM modules. New modules to support LU6.2 processing supplied on release tape only to those who have purchased the feature (basic (passive mode) and/or extended (active mode) versions).
- To allow local copy processing to a printer, an 'End Bracket' command is added to the last transmission if no more messages are queued.
- VTIDTABL - VTAM-id table can be generated internally (instead of coding VTIDTAB macros) by coding the VTID parameter on all applicable LUNIT macros (ICOM-id/VTAM-id cross-reference indexes also generated internally).
- VCT macro, APPLID (and PASSWD) parameter - can be overridden via a new EXEC statement parm value (APPLID=name), or the VTCN\$START command (APPLID parameter), thus allowing the same VCT coding (Network Table) to be used for multiple Intercomm on-line systems.
- COPYEXIT - user exit (sample supplied) can also be used for VTAM processing (see Section 2.4.1).
- Back End RLSE (FECMRLSE) processing - request ignored if a message is already on the queue (FECMRLSE must be queued before message to be released, or use FESEND R option on previous message).
- FLSH command - processing modified to eliminate 'chaser' message (queuing problems) and to allow flushing one, n, or ALL messages on the queue (count stored in VTAMSCTS queue entry for LUC), and optionally to flush all messages on disconnected (down) originating LUCs whose alternate terminal field points to the LUC for which the FLSH command was entered.
- Over 10,000 terminal definitions are now supported via internal table macro changes. External changes to be made by the user are described in the installation chapter of the SNA Terminal Support Guide. Assembly and linkedit (of Network Table) recommendations are also described.
- DDQ output transmission error recovery - DDQ transmissions are logged with new log codes (F5-F8) showing header and text. When transmitting, a response is requested so that an erroneous message can be flushed and transmission can continue (instead of retransmitting from the beginning of the DDQ (occurs only if session lost)). The user can examine the log records to determine erroneous data in flushed messages.
- Session Manager support has been added (VCT - SESSMNGR parameter).
- CPU Console may be defined as a VTAM LUNIT (does not have to be the control terminal).

- F3 log records - new use of MSGHUSR field: an F indicates the message was flushed internally or by FLSH command, a Z indicates message rejected (thrown away) by OTQUEUE user exit.
- VTAM node or region failure recovery - code added to prevent loss of a large VTAM node, or the VTAM region, from abending Intercomm.
- VCT macro, new MDOMAIN parameter (YES/NO) - indicates whether LOGMODE options (VTLSEB macro) should be ignored for a cross-domain LOGON.
- VTCSB/LUNIT/LCOMP macros, new NORLSEM parameter (YES/NO) - for 3270 CRTs: suppress output text response to RLSE command, just send keyboard reset command (WCC).
- New command options, display options, etc. - see Section 2.15 on Command Changes and Additions.
- Size of RPL pool acquired at startup reduced to 1 (instead of 2) per LUNIT or per SNMAX value (if defined), plus RPLs for RECEIVE processing.
- VCT has new parameters to control snap production (FULLSNAP), WTOs/traces/snaps at closedown (CLOSE), and control terminal acquisition when a VTAM LU (CONTROL).
- Both Intercomm tid and VTAM-id given in LOGON and LOGOFF messages.
- SIMLOGON (internal or external STLU\$ACQ\$Q) not queued if LU inactive (queued only if current owner has a RELREQ exit that can be posted or LU is not at session limit defined to VTAM) and STLU is rejected.
- If SIMLOGON fails, Shared Printer messages automatically routed to alternate (if defined or assigned and in session).
- 'Dynamic Screen Size Log Mode' is recognized for 3270-type CRTs and the default screen size of 24x80 is assigned.
- SSPOLL supports VTAM (suspend new RECEIVES if Cushion is released).
- VTAUTOUP also called, if VTUPINV defined on VCT, after internal VTCN\$HALT issued.
- VXQCB chaining by VTEXTS now FIFO for debugging VTAM exit activity.

Documentation Changes: SNA Terminal Support Guide
System Control Commands
Messages & Codes

New External Options: see above.

Installation: Automatic.

2.10 MESSAGE MAPPING UTILITIES (MMU) CHANGES

The following enhancements have been made for MMU processing and control:

- Map Group assembly - internally add system date and time stamp in new fields (generated via ENDGROUP macro).
- MMUC command - new DATE subcommand to display last assembly date and time (if available).
- LMAP command - new command to dynamically load or reload a Map Group load module to the on-line maps Store/Fetch data set. New LMAP subsystem is Link Pack eligible.
- MAPOUT processing - new V option code (byte 4 of MCW) requests that only symbolic map data (if any) and/or attributes overrides (if any) be sent in the output message (no initial data or attributes from map to be used). A WRITE command (X'F1') is used automatically, however, the user may need to use 'RKEYBD' CNTLCHR WCC override (to reset keyboard and not reset MDTs) for the MAPEND call.
- MAPOUT processing - a return code of 73 is supplied in first 2 bytes of MCW if the specified output tid is incorrect (not found in Station Table), instead of forcing a program check.
- New MAPEND processing options added:
 - 1) R - same as Q, but call FESEND with 'Release Next' option
 - 2) 'copies' parameter when option D used - queue up to 255 copies of output from a DDQ to a printer (multiple report copies produced).

Documentation Changes: Message Mapping Utilities
System Control Commands

New External Options: MMUC command: DATE subcommand.
 LMAP command.
 MAPOUT call: V code (byte 4 of MCW).
 MAPEND call: R code (byte 2 of MCW)
 'copies' parameter with D code.

Installation: Automatic.
 For LMAP command, add DD statement(s) for map load library (MODMDF) to STEPLIB concatenation.
 Ensure SYCTTBL defined for LMAP subsystem (provided via INTSCT, etc.), LMAP in linkedit, LMAP verb in BTVRBTB (use supplied version), and RPT00028 (for output responses) in linkedit.
 Ensure subsystem codes (SUBH=L,SUBC=M) defined for LMAP for applicable Satellite Regions in PMIRDT.
 If COBOL used, recopy, or regenerate LOGCHARS copy member COBLOGCH to user copy library.

2.11 VSAM SUPPORT ENHANCEMENTS

The following enhancements have been implemented for VSAM file support:

- ESDS empty file loading supported, as well as writing over an existing file.
- Data set name sharing for Paths supported.
- New buffer sizes (MVS/XA 2.2 - DFP V2.3 and up) supported for LSR pool.
- LSR pool may be created above the 16M line.
- LSR pool stats for odd CI sizes and for index CI buffers also collected (added to nearest higher size).
- If a user thread needs to wait on access to a VSAM CI (no buffer, STRNO exceeded, CI in exclusive control), it is enabled (to allow purge if thread times out) before a 1/3-second interval wait is issued, and then disabled again (to prevent purge) before a retry is attempted. During the wait interval, a TALY\$DA will show the thread status as WAIT-CI. ENABLE/DISABLE is not issued if Backout-on-the-Fly is active for the thread and the data set (access must complete), but wait (and CI status) is done.
- VSAMCRS (FAR option) processing corrected and need for SVC support for this option has been eliminated.
- VSAM files supported for Dynamic File Allocation special feature ACCESS calls (see Section 2.12).

2.11.1 ESDS File Loading Support

Previously, to load a preallocated ESDS file on-line under Intercomm, it was necessary to have at least one record on the file because it was opened for input-output processing. Now, if an ESDS file is empty, it will be opened for sequential output processing (PUTV ADD), and the SELECT return code is 1. No GETV against the file may be issued until it is closed after one or more records are written. If the file has existing records, it is opened for input/output (SELECT return code is 0), and PUTV ADD will add records to the end of the file. However, if a FAR is specified with the WRITEOVER option for the file (and the REUSE attribute was specified on the DEFINE statement when the file was created via IDCAMS), then the file is treated as though it is empty and PUTV ADD requests will overlay the existing records (file opened for output only). The LSR option may not be requested for empty or reused ESDS files. The user must ensure single threading of PUTV ADD requests.

Note that the file status may be changed on-line via the FILE command, ALTER subcommand. If the WRITEOVER FAR was omitted, a new ALTER\$WRITEOVER option is available (file closed and reopened for output only). To change the overwrite processing to allow GETV processing, use the ALTER\$WRITES option.

Documentation Changes: Operating Reference Manual
 Programmers Guides
 System Control Commands

New External Options: FAR: WRITEOVER option.
 FILE command: ALTER\$WRITEOVER option.

Installation: Automatic.

2.11.2 Path Data Set Name Sharing Support

If VSAM KSDS records are accessed and/or updated via one or more paths and/or a path and the base cluster, file integrity (accessing the latest version of a record) is only guaranteed by using the VSAM 'Data Set Name Sharing' option, even if the path(s) and base cluster are all connected to the LSR pool. To implement this option, a new FAR option, DSN, must be specified for the path(s) and base cluster files. This option requires that LSR buffer pools be implemented for the files and internally forces connection to the pools, even if LSR omitted on the FARs. The FAR for the base cluster must be defined before that for the path(s). Related Alternate Index DD statements may not be in the Intercomm JCL. UPGRADE and UPDATE parameters are required on the DEFINE statements when the files were created via IDCAMS.

Documentation Changes: Operating Reference Manual
 Programmers Guides

New External Options: FAR: DSN option.

Installation: Automatic.

2.11.3 New LSR Buffer Pool Sizes

Under MVS/XA 2.2.0 and up, using DFP V2.3 or higher or if using DFSMS, the following CI and LSR buffer pool sizes may be specified: from 512 to 8192 (8K) in increments of 512 bytes, and from 10240 (10K) to 32768 (32K) in increments of 2048 (2K). These sizes are supported for the BUFFERS parameter on the SPALIST macro to specify the LSR pool, and for statistics gathering and display.

Documentation Changes: Operating Reference Manual
 Basic System Macros

New External Options: SPALIST macro: BUFFERS parameter (modified).

Installation: Automatic.

2.11.4 LSR Pool Above the 16M Line

A new parameter (LSR31=YES/NO) has been added to the SPALIST macro to indicate whether the VSAM LSR pool and related control blocks should be built above the 16M line to save storage in the Intercomm address space (requires DFP V2.4 or higher, or DFSMS). A new message (FR105I) indicates at startup (if LSR specified or forced for any VSAM files) where the pool was built. By default, the pool is built below the 16M line if the BUFFERS, etc. parameters are specified on the SPALIST macro.

Documentation Changes: Basic System Macros
 Operating Reference Manual

New External Options: SPALIST macro, LSR31 parameter.

Installation: add LSR31=YES to other LSR pool parameters.

2.12 DYNAMIC FILE ALLOCATION SPECIAL FEATURE MODIFICATIONS

The following changes to this feature have been made:

- DFA support code (ACCESS and ALLOCATE entries) have been moved from the File Handler (IXFHND01) to a new module containing those entries (IXFDYNAM).
- DYNALOC global removed from INTGLOBE and SETGLOBE.
- IXFDYNAM is Link Pack eligible (code DFA option on LPSPA and LPINTFC macros).
- Installation and linkedit supported via DFA (YES/NO) parameter on ICOMGEN and ICOMLINK.
- Support added to IXFDYNAM for catalogued VSAM files (ACCESS calls only): code V in byte 3 of FHCW, code R in byte 2 if READONLY processing to be done. Standard GETV and PUTV calls then used to access records.
- DISP=SHR supported for VSAM files (code S in byte 2 of DDSASTAT).
- After call to ACCESS for a VSAM file, CI size returned in new DDSACINV field in DFACB.
- Empty VSAM ESDS data sets may be accessed and then loaded (PUTV ADD) if the data set is predefined and catalogued via IDCAMS. (Recognition of the empty file for OPEN processing executed internally.)

Documentation Changes: Dynamic File Allocation
 Operating Reference Manual
 Basic System Macros
 Installation Guide

New External Options: VSAM file support via ACCESS (C'V' in byte 3 of FHCW);
 read only supported (C'R' in byte 2 of FHCW).
 DISP=SHR for VSAM files (C'S' in byte 2 of DDSASTAT).
 LPSPA, LPINTFC macros: DFA option.
 ICOMLINK macro: DFA parameter.
 ICOMGEN macro: DFA parameter.

Installation: Automatic (if DFA=YES coded on ICOMGEN, ICOMLINK),
 else ensure IXFDYNAM included in Intercomm linkedit.

2.13 TOTAL DATA BASE SUPPORT ENHANCEMENTS

Several enhancements have been made for TOTAL support:

- CONTIN (YES/NO) parameter supplied for TOTFLGEN file definition macro so that more than 50 files for individual file definition parameters (READ, IUPD, SUPD, and/or EUPD) can be defined.
- TOTSTART modified so informative messages (DB103I) issued for each file for which an OPEN failed (if any), and only one WTOR (DB700R) issued at end after all open processing, if one or more opens failed (count given in message). This saves operator replies for each failure when only a partial data base used.
- PDATBASE - stop retries for a HELD or TFUL condition if thread has timed out (allow thread purge).
- PDATBASE - set data-base-access-in-progress flag for TALY\$DA command.
- Implement an automated TOTAL restart coordinated with Automated Message Restart facility. This feature uses five TOTAL logs coordinated with the five Intercomm checkpoints (new log started after each checkpoint taken), a supplied TOTAL user exit (CSITA014) to do the logging, a TOTAREOF utility (execute before TOTRSEX recovery program) to attach ICOMFEOF to process the last TOTAL log when restart needed, and a TOTLOGSW data set to provide previous and current TOTAL log names (initialized by AUTORSET program).

Documentation Changes: DBMS Users Guide (SPR 232 for TOTAL)

New External Options: TOTFLGEN macro: CONTIN parameter.

Installation: see documentation.

2.14 EXTENDED SECURITY SYSTEM ENHANCEMENTS

The following enhancements have been implemented:

- Sort in-core account list at startup - use binary search at sign-on
- Sort exempt terminal list at startup and when modified (for display)
- Set off Global and Manager attributes in end user records
- Display user resource lists (verbs, terminals, regions) in sorted order
- Display user list, signed-on user list, exempt-tid list in sorted order; display partial list by generic name or group name
- Allow a Group Manager to attach individual resources from his list to end-users in his group
- Optimize delete processing (back chain of user records added) so other ESS processing allowed; log delete transaction under ESS log code
- Provide ATTACH/DETACH,user-id,ALL option - copy all lists from Group Manager/user to end-user within the same group
- Add DISPLAY,DEACT option - list user accounts which cannot sign on
- Allow Group Manager to attach/detach resources/list or modify attributes for the entire group or generic user-id within the group, in one command
- Allow Global Manager to attach/detach resources or modify accounts using a group name, generic group name, or generic user-id, in one command
- Add DISPLAY\$DEFAULTS to display Default Attribute List (requires Global authority)
- New AID attribute (VTAM) - set AID group number in LUC at sign-on (value coded for LUNIT's VTCSB is default), add to account display
- Add/display account last modify date, usage total, password change date
- Log account ADD/ATTACH/DETACH/MODIFY transactions
- Log Default Attribute List modification
- Log 'User-id is currently active' response to sign-on
- Encrypt password on Security file
- Hardcopy - cover user-id and password entry area for sign-on

- SECUPRNT - new off-line utility to print all user accounts or optionally print for a specific or generic group name; optionally also request record chaining and resource count validation. Also prints control record values, news records (if any), exempt tid list, and default attributes (all lists and user records printed in sorted order if SECUPRNT is linked with INTSORT)
- Supplied User Exits - SECUEXIT (signon LOGOFF user-id check, signoff cleanup); LUCUR (VTAM LU HALT exit - force signoff); USRPAGEX (reject adding Page Facility pages if user signed off)
- Call SECUEXIT (code = 16) for Time-out (note - time-out logged even if exit returns non-zero code (signoff forced via SPLU in user exit))
- User identifying data may be added to user account record when created, or via MODIFY command, which will be displayed on account profile screen, SECUPRNT report
- CPW (change password) command added (user forced off to do change)
- Password expire (PSWDEXP) attribute added to specify maximum number of signons before password change required
- MODIFY\$PASSWORD and CPW command requests also passed to the SECUEXIT user exit for checking/modifying account expire date (address of user account record passed as a third parameter for all options except signoff and timeout)
- SEND command can be used by a Group Manager to send a message to all signed-on users in the group, or by a Global Manager to all (or a group or generic subset) signed-on users
- FORCE command can specify a group name or generic user-id subset
- The new Integrity SVC (IISVC) supports ESS processing requests for protected storage, modifying SECVECT
- SECUPTRS utility created to back chain user records (for delete processing) in an existing file, or to print user account-ids with chain pointers, and/or control record values (chain pointers)
- Invalid resource lists are checked for, reported, if count/record-chaining bad
- Queuing of signon screen can be suppressed to save storage if LOGOFF occurs before signoff (requires LOGON exit to internally return signon screen: sample available).

Documentation Changes: Extended Security System

New External Options: See above.

Installation: see Extended Security System.

2.15 SYSTEM CONTROL COMMAND ENHANCEMENTS

The following system control commands have been revised or added:

- LMAP - new command to reload a Map Group to MMU S/F data set
- MMUC - new DATE option to display last assembly date, time
- LOAD - new GET sub-option to change DWS/ISA size for reloaded COBOL or PL/1 subsystem, or reentrant COBOL subroutine
 - CORE option increased to 9999K (9+ Meg) below the 16M line
- SPLU - new ATD sub-option to reroute output messages
 - new PASS (to another APPLID) option
- VTST - VTxxxxxxx parameter to display specific LU data using VTAM-id
 - display LUB active status
 - provide generic tid/VTAM-id display generation (in alphabetic order)
 - new SIM option - list terminals with SIMLOGON queued
 - new ALU6 parameter to list only LU6.2 LUs
 - new REL option - display connected LUs with RELREQ pending
 - display max sessions allowed on totals line
- WHOI - new AID key assignments display (entering terminal)
- WHOU - new AID key assignments display (another terminal)
- BRUP - STLU a group of terminals based on generic lcom-id
- BRDN - SPLU a group of terminals based on generic lcom-id
- VTUP - STLU a group of terminals based on generic VTAM-id
- VTDN - SPLU a group of terminals based on generic VTAM-id
- BTUP - TPUP a group of BTAM/TCAM terminals based on generic lcom-id
- BTDN - TDWN a group of BTAM/TCAM terminals based on generic lcom-id
- STAT - add VERB and REGION if terminal locked
 - display requested line/tid if not found
- LTRC - dial-up line traces by terminal only, spin off BTAM snaps
- RLSE - new NORLSEM parm (BDEVICE/CSB/LUNIT) - reset keyboard only
- FLSH - for BTAM/TCAM if dedicated queues and VTAM (all devices)
 - eliminate chaser message
 - allow flush 1, n, or ALL queued messages
- QHLD/QRLS - supported for VTAM (printers)
- STLU/SPLU/RSLU - VT (VTAM-id) subparameter added - can be used instead of TPU,
 - also added to LOCK/UNLK/FLSH/QHLD/QRLS/TPUP/TDWN

- VTCH** - APPLID (and PASSWD) option to define Intercomm VTAM-id for START
- QUICK option on HALT parameter (immediately close VTAM Front End)
 - SNMAX parameter to change maximum concurrent sessions
 - SNPOF/SNPON parameters to suppress/allow ID=63 (full region) snaps
 - TRCOF/TRCON parameters to suppress/allow ID=26 error snaps
 - CLSOF/CLSON parameters to suppress/allow at VTAM closedown:
 - error trace (ID=26) or error indicator (ID=61/62/63) snaps, or
 - VTAM error messages (WTO's)
- COPY** - add a Form-Feed (FF) to output to a VTAM printer
- LOKA** - new command to lock a verb to a remote APPLID (extended LU6.2)
- ULKA** - new command to unlock a verb from a remote APPLID (extended LU6.2)
- COMM** - allow 3-digit decimal number for subsystem codes
- FILE** - add a STAT function to display:
- file type
 - current status (open/closed/deallocated/locked)
 - access option (read only, update, write only)
 - FAR options (LSR, etc.)
 - file recovery options
 - number of current SELECTs
 - DISP (NEW/MOD or OLD/SHR)
- add a WRITEONLY sub-option to ALTER function
- provide deallocate/automatic reallocate SMLOG, SYSPRINT
- display data for new buffer sizes under LSR option
- SCTL** - new command to:
- display core, Intercomm control blocks, modules
 - provide thread dump for a specific thread number
 - provide a display of a specific WQE queue trace
 - generate dynamic call to TDUMP or IJKTRACE
 - modify STOCORE, MMNCL, MDELY, SNAPPGS, PFMAXS
 - locate address from Csect/entry-point/load module name, or
 - name from address (above/below 16M line)
- STRT/STOP** - indicative dumps by id (114, 118, 126) - also specify in SPA
- TLOOP (IJKTLOOP processing)
 - CB2SNAP (VS COBOL II snaps)
 - POOLDUMP (Intercomm core pools printout after a program check)
 - TABSNAP (Table Facility snaps)
- PAGE** - new response termination options (TC/TH/TL)
- new E option (display last page of all responses)

- TALY - SU - messages completed/queued count
- core in use by Store/Fetch strings/max core allowed/flush count
 - core in use by loaded subsystems/max core allowed below 16M line
 - number of times MAXLOAD reached (cleared after LOAD\$CORE issued)
 - count of log buffer waits
 - count of WAITs issued by Dispatcher
 - Table Facility usage
 - count of current/minimum WQE's on FREE Queue
 - count of RCB Table relocations
 - Region name on time stamp line
 - current/high thread count on time stamp line
- DA - add input message terminal-id
- add WAIT or HUNG reason code (I/O, DB, NQ, CI, etc.)
 - indicate if subsystem loaded above 16M line
- NG - list all non-schedulable subsystems
- RH - reset thread high count to current number
- BE - add current subsystem PRTY level
- MU - list all subsystems with non-0 MAX USAGE (processing MNCL threads)
- NQ - list all subsystems with queued messages
- NC - list all subsystems with cancelled messages
- NS - list all subsystems with started messages
- NP - list all substems which have processed messages
- B* - list all subsystems where any above count non-zero
- FE - modify for LU6.2 session info
- FN - list only terminals without queued messages
- FQ - list only terminals with queued messages
- FU - list up/logged-on terms with Q'd messages
- FD - list down/logged-off terminals with Q'd messages
- FI - list deactivated LUs with Q'd messages
- BD - list only down BTAM/TCAM terminals
- BU - list only up BTAM/TCAM terminals
- VS - list only in-session VTAM terminals
- VN - list only logged-off VTAM terminals
- VI - list only deactivated VTAM terminals
- FE(tid) - list only for specified Intercomm terminal-id
- BE(ssch\$ssc) - list only for subsystem specified via codes ssch\$ssc

Subsystem processing control Fine Tuner commands:

MNCL - resets MAX USAGE count to zero when MNCL changed

DELY - reject input messages while delayed (see SYCTTBL macro changes)

FTUN - new command to display or reset subsystem SYCTTBL parms (uses MMU):

- logging (yes/no),
- TCTV time, PRTY, GET/SPAC, MNCL values,
- SAM, BACKOUT, RVFILE (yes/no),
- COBOL DWS checking (yes/no),
- reverse CANCEL=STOP,
- indicative dump processing (yes/no),
- reverse SNAP 118 dumping after time-out (yes/no),
- stop subsystem processing (reject input messages),
- display Queue status, messages counts,
- display loaded status, number of loads

SSUP - new command (in second FTUN screen) to modify SYCTTBL fields

TCTV - new command to change subsystem time-out value

SSFL - new command to flush 1, n, or ALL messages queued for a subsystem

SPAC - new command to change DWS/ISA area size (GET/SPAC parms)

PRTY - new command to change subsystem execution priority.

Documentation Changes: System Control Commands

New External Options: see above.

Installation: Automatic, except as noted in System Control Commands - Appendix A.



1. 2.

1.



2.



Chapter 3

NEW DOCUMENTATION

All of the Intercomm documentation is being updated to correspond to Release 10, where applicable. The following manuals have or will have SPRs or new editions:

- ASMF Users Guide (SPR 239)
- Assembler Language Programmers Guide -- New Edition - 6/89
- Basic System Macros -- New Edition (plus SPRs 241, 245, 247)
- BTAM Terminal Support Guide (SPR 236)
- COBOL Programmers Guide -- New Edition - 4/94
- Concepts and Facilities (SPR 230)
- DBMS Users Guide (SPR 232)
- Dynamic Data Queuing Facility (SPR 231)
- Dynamic File Allocation -- New Edition - 7/87
- Extended Security System -- New Edition - 7/92
- File Recovery Users Guide (SPR 237)
- Installation Guide -- New Edition - 2/95 (for SM Level 2300)
- Messages and Codes -- New Edition - 3/94
- Message Mapping Utilities (SPR 246)
- Multiregion Support Facility -- New Edition
- Operating Reference Manual -- New Edition (for base Release 10)
- Page Facility -- New Edition - 5/94
- PL/1 Programmers Guide -- New Edition - 2/91
- Planning Guide -- New Edition - 1/95 (for SM Level 2300)
- SNA LU6.2 Support Guide -- New Edition - 2/93
- SNA Terminal Support Guide -- New Edition - 7/93

- Store/Fetch Facility -- New Edition - 2/94
- System Control Commands -- New Edition (plus SPR 242)
- Table Facility -- New Edition (new facility) - 8/93
- TCAM Support Users Guide (SPR 238)
- Technical Information Bulletins --New Edition
- Utilities Users Guide -- New Edition

Some of these updates will be mailed with the Release 10 installation tape. The rest have been, or will be, published when ready (see the latest Publications Price List/Order Form). The SPR to the ASMF manual was mailed with an SM tape. New/revised indexes to the above manuals are being prepared, where necessary.

Appendix A

INTERCOMM MODULES ADDED/DELETED/REVISED FOR RELEASE 10

Added:

VERBSTRT	IXFDYNAM	DDQPRT (Utility)
INTSORT	DWSSNAP	CORACCT (Dsect)
COREACCT (Macro)	VT62DATA (Dsect)	VTBIND62 (Macro)
VTCDM6	VTLUDM6	GETDATE (Macro)
QCOUNT (Macro)	SECUPRNT (Utility)	SSRPT
SUBRPT	DYNSSUP	SYSCNTL
LMAP	CICSSAMP (Sample Program)	FEWHOI
SPLEVEL (IBM Macro)	XASWITCH (Macro)	AUTORSET (Utility)
AUTORCVR	LOGMERGE (Utility)	TOTAREOF (Utility)
STUSWRCD (Macro)	USRPAGEX (User Exit)	USRQMONX (User Exit)
USERB37E (User Exit)	COPYEXIT (User Exit)	SECUEXIT (User Exit)
LUCUR (User Exit)	CSITA014 (TOTAL Exit)	MYPRTTAB (Table)
SWMODE	USRSTSCH (User Exit)	SECUPTRS (Utility)
STVSCOB2	DUMVCOB2	CB2SNPAR (Dsect)
THDCOM (Dsect)	INTTABLE	INTTABDS (Dsect)
IGCICSVC (IISVC)	GETDLU (Macro)	PULLDLU (Macro)
PUTDLU (Macro)	VTCDM62	VTLUDM62
INITLU6	VTPASS62	

Deleted:

CKOVLYNO	PMIFIXA	ICOMFIX	PMIFIXB
FIXSECT	INTPAGEC	INTPAGE	VS1LOADR
VSINIT	AMGINTFC	CFMSINTF	GAMFQES
FQES	TUNRTBLC	DISCONV	DISREORG
IXFDISAM	GENESIS3	TRAFFICQ	PMIDCB
IGCFAST	MMUDDMF	GRAPHICS	BUNKRAMO
PMIWILT	IGG019U3	IGG019WK	UNVTRST
WILTEKAE	WILTMULT	WILTOPEN	LOCDSECT
PAGETBL	PAGETBLE	SRCHPTBL	DYNLINK
IFCICS	SENDSECT	Assembler F and PL/1-F Procs	
PMIRJE	RJ..... (OS RJE Special Feature modules)		

Revised/Rewritten/Resequenced:

SFTABLE	SCTLISTC	PMISERC3	SYCT400
TALLY	FECMD	VTAMSTAT	FINTUNER
TUNRBEGN	RPT00008	RPT00009	RPT00028
RPT00043	RPT00045	ABTOTEND	DYNDSECT
IXFCTRL	BLHOT	VTLUCMD	VTEXTS
INTLOAD	INTTIME	IJKCESD	IJKWHOIT
LOADSCT	ICOMDYNL	ICOMCESD	ICOMVCON
RQEDSECT	PGEDSECT	PAGE	PAGEMSG
KEYFLIP	MSGHDRC		

Note: Many others have changes but were not resequenced. User mods to all modules must be carefully evaluated for applicability and necessity.

Appendix B

New Intercomm Message Header Layout

Field Name	Length	Description	Alter Legend
MSGHLEN	2	Length of message including header (binary number)	Y
MSGHQPR	1	Teleprocessing segment I/O code: 02/F2=full message; 00/F0=header segment; 01/F1=intermediate segment; 03/F3=final (trailer) segment	N
MSGHRSCH	1	High-order receiving subsystem code	Y
MSGHRSC	1	Low-order receiving subsystem code	Y
MSGHSSC	1	Low-order sending subsystem code	M
MSGHMMN	3	Monitor message number assigned by Message Collection (binary)	N
MSGHDAT	6	Julian date (YY.DDD)	N
MSGHTIM	8	Time stamp (HHMMSSTH)	N
MSGHTID	5	Terminal identification (originating terminal on input messages/ destination terminal on output) or Broadcast Group name	Y
MSGHFLGS	2	Message indicator flags (MSGHCON - R9)	N
(MSGHPID)	2	Reserved area (MSGHFLGS - R9)	N
MSGHBMN	3	Front End message number (binary)	N
MSGHSSCH	1	High-order sending subsystem code	M
MSGHUSR	1	User/system processing code	L
	2	Reserved for special processing by the Front End (MSGHBMN - R9)	N
MSGHLOG	1	Log code	L
MSGHBLK/ MSGHMRDX/ MSGHRETN	1	Reserved area/ Multiregion region number (log code 01/30)/ Subsystem return code (log code FA/FD)	N
MSGHVMI	1	Verb or message identifier interpreted by receiving subsystem as required, and by FESEND	Y

