INTERCOMM

BASIC SYSTEM MACROS

ISOGON
CORPORATION
330 Seventh Avenue, New York, New York 10001

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 derivitave 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.

Publishing History

Publication	Date	Remarks
First Edition	September 1973	A new manual titled System Macros Manual.
Second Edition	September 1986	Incorporating Release 9.0 changes and updates for Release 10.

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

PREFACE

Intercomm is a state-of-the-art teleprocessing monitor system executing on the IBM System 360/370 family of computers and operating under the control of IBM Operating Systems (MFT, MVT, VS1, MVS). 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.

This document describes user-applicable Intercomm macro instructions, and is intended for use as a reference by all system and application programmers associated with the Intercomm system. As a reference manual, only specifications for coding are given. In conjunction with the use of this document, the reader should consult the following Intercomm publications:

- Assembler Language Programmers Guide
- Operating Reference Manual
- BTAM Terminal Support Guide
- Generalized Front End Facility
- TCAM Support Users Guide

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

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

- Amigos Users Guide
- 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
- Remote Job Entry (OS)
- Store/Fetch Facility
- SNA Terminal Support Guide
- TCAM Support Users Guide
- Utilities Users Guide

TABLE OF CONTENTS

The following list is a locator device for all Intercomm user-coded macros. Where a page number appears, the macro is contained within this volume. Where a code, such as MMU, MRS, etc., appears, the macros are to be found within the indicated manual. The following abbreviations are used:

ASM = ASMF Users Guide

DBM = Data Base Management System Users Guide

DDQ = Dynamic Data Queuing Facility

ESS = Extended Security System

GFE = Generalized Front End Facility

MMU = Message Mapping Utilities

MRS = Multiregion Support Facility

MSG = Model System Generator

PAG = PAGE Facility

SNA = SNA Terminal Support Guide

UUG = Utilities Users Guide

MACRO CODING CONVENTIONS

Each macro description is accompanied by a form illustration. This illustration designates which operands are required, which are optional, which must be coded exactly as shown, which may be iterated, etc. The conventions for the presentation of the material in these illustrations are as follows:

- A keyword operand is presented in uppercase letters followed by an equal sign. (For example, TSTEND= on SPALIST macro.)
- A positional operand is represented by a name in lowercase letters; it is never to be coded, but is always to be replaced by a permissable expression. (For example, "list-type" on POLLIST macro.)
- A code element consisting solely of uppercase letters represents already encoded information; it must be written exactly as shown. (For example, LIBR=SYSLIB on ICOMLINK.)
- A code element consisting solely of lowercase letters represents information not yet encoded; it is to be supplied in encoded form by the programmer. (For example, LIBR=ddname on ICOMLINK.)
- All punctuation symbols are to be coded exactly as shown.
- ,... An elipsis indicates that multiple iterations of an operand may be specified.
- { } A pair of braces discloses the presence of a required choice: code elements contained within the braces represent alternatives, one of which must be chosen. The braces are not to be coded.
- [ ] A pair of brackets discloses the presence of an optional parameter or subparameter: code elements contained within the brackets represent alternatives, one of which may be chosen. The brackets are not to be coded.
- { NO / YES } An underlined code element indicates the default code, if the associated parameter is omitted.
- (r) Parameters may provide for or require general register usage, (r), to contain a value or address. Often, (0), (1), (14) and (15), listed with these parameters, designate specific registers to contain the parameter value. If these listed registers are used and initialized as expected, unnecessary instructions will not be generated, thereby reducing execution time and storage requirements.

- [symbol] A pair of brackets enclosing the lowercase word "symbol" in the label field of an illustration discloses the permissability of the presence of any symbol valid in the Assembler Language.

- (blank) Parenthesis enclosing the lowercase word "blank" in the label field means the field should be left blank, as the macro instruction generates its own symbol.

- In the operand field of the illustrations, a set of one or more lowercase words followed by a colon is a heading descriptive of one or more subsequently illustrated parameters; for example,

  leased line parameters:

  in the BLINE illustration.

- In the description of macro parameters, all references to value ranges are references to inclusive ranges.

- Any reference to a character or bit string is a reference to a connected sequence of characters that is treated as a coded unit.

- All numeric fields should specify significant digits only. (For example, SPALIST macro, LGNUM parameter, specification "#-of-interlog-buffers".)

INTRODUCTION

This publication contains the specifications for the basic system macro instructions for use with Intercomm. They are arranged in alphabetical order, with the macro name appearing in the top corner of each page.

There are essentially two kinds of macros:

- Those whose assembled code is executable
- Those whose assembled code is formalized into one or more table entries

The presentation herein pertains solely to coding specifications for the macro instructions. Operation and implementation considerations of the Intercomm system are described elsewhere.

All Intercomm macros remain compatible from release to release. Obsolete parameters or specifications have been omitted from this manual. If coded, they will be ignored.

Macro instructions for Special Feature products are located within the related manual. For example, macros used to construct tables or define specifications for the Change/Display, Edit and Output Utilities are documented in the Intercomm Utilities Users Guide. Macros required in implementing the Message Mapping Utilities (MMU) are documented in Message Mapping Utilities.

The Listing of Macros in the Table of Contents of this document is a complete locator-index of all Intercomm system macros available to the user.

NOTE: Within various Intercomm source modules, the user will encounter macros which are for internal use only. These macros are not documented in this manual. Certain user-coded macros, such as the SPALIST macro, contain parameters which are for internal use only or are obsolete (unsupported). In this case, user-coded parameters are documented and internal or obsolete parameters are not.

AIDDATA -- Define 3270 Attention Identification Data String

The AIDDATA macro defines a data string to be used as a prefix to input from IBM 3270 terminal(s). In order to be utilized as such, it must be referenced by an attention operand of an AIDGRP macro. All AIDDATA macros must be coded together, because they exist in a CSECT named AIDDATA.

The form of the AIDDATA macro is as follows:

symbol	AIDDATA	{num} {END} [,'character-string'] [,REPLACE={YES}] [{NO }]

num
: defines the number of this AIDDATA macro. This is referenced by one or more AIDGRP macros. Code as a decimal number in the range 1 to 255.

END
: is coded as shown, with no other operands, following the last numbered AIDDATA macro. This format does not define a data string, but properly terminates the AIDDATA Table, including generation of a PMISTOP macro.

'character-string'
: provides the character string prefix (delimited by single quotes). The user may add data to the string by coding DC instruction(s) immediately following this AIDDATA macro. If 'character-string' is omitted, the entire data string is assumed to be coded by the user with DC instructions which may specify hexadecimal and/or character data.

REPLACE
: specifies whether the coded 'character-string' and/or subsequent DC instruction(s) is to completely replace data received from the terminal when it accompanies a PF key AID value. Do not code a message ending character (X'26' or X'37') at the end of the data string. No data is transmitted when a PA or the CLEAR key is used. The default is NO, indicating that the coded string/DCs are to prefix any data received from the terminal.

AIDGRP -- Define a 3270 Attention Identification Character Group

The AIDGRP macro corresponds each of a group of IBM 3270 terminal Attention Identification Characters with a data string in the AIDDATA Table, that is, an AIDDATA macro. In order to have the data string accessed, the AIDGRP macro must, in turn, be associated with a BDEVICE or LUNIT/LCOMP or VTCSB macro for one or more IBM 3270 terminal(s).

The form of the AIDGRP macro is as follows:

symbol	AIDGRP	num,attention=a[,attention=a,...]

attention=a

equates a 3270 Attention Identification value with an AIDDATA macro. "a" is the 'num' value of the corresponding AIDDATA macro and must be decimal in the range of 1 to 255. 'attention' must be one of the following keywords:

PA1, PA2, PA3 -- Program Access Keys

or

PF1, PF2, PF3, PF4, PF5, PF6, PF7, PF8, PF9, PF10, PF11, PF12, PF13, PF14, PF15, PF16, PF17, PF18, PF19, PF20, PF21, PF22, PF23, PF24 -- Program Function Keys

or

CLEAR -- Clear Key

or

PEN -- Selector Pen Input

or

CARD -- Badge Reader Input

num

defines the number of this AIDGRP macro. This is the same as the AIDGRP operand of the corresponding BDEVICE, LUNIT/LCOMP or VTCSB macro. Code as a decimal value of 1 to 255.

BCGROUP -- Define a Broadcast Group

The BCGROUP macro is used to define a broadcast group within the Intercomm Back End Broadcast Table (PMIBROAD).

The form of the BCGROUP macro is as follows:

(symbol)	BCGROUP	GROUP=group-name, TERMS={term } {(term,...,term)} [,CONTIN={YES}] [{NO }]

CONTIN
: specifies whether the remainder of a group list can be found in succeeding BCGROUP macros. This is necessary whenever the terms parameter exceeds the Assembler limit for a macro instruction parameter (255 characters, including commas and parentheses). A BCGROUP macro with CONTIN=NO completes a group. The default is NO. A maximum of 32,767 terminal ids may thus be defined.

NOTE: The GROUP parameter (and comma) may be omitted on all but the first BCGROUP macro in a group.

GROUP
: specifies the one- to five-character name of the broadcast group.

TERMS
: specifies one, or a list of terminal identifiers (five characters each, as coded for the STATION macro) of terminals in the broadcast group. Enclose the list in parentheses, each terminal-ID separated by a comma.

BDEVICE -- Define Device (BTAM/TCAM/GFE Front End)

The BDEVICE macro is used in conjunction with the LINEGRP, BLINE and BTERM macros to generate the Front End Network Table. The BDEVICE macro is used to specify information concerning a single device type. One or more terminals are associated with a device type via identical codes assigned to their representative BTERM macro, DEVIND parameter. Refer to the BTAM Terminal Support Guide for additional information on specific device types.

The form of the BDEVICE macro is as follows:

```
(symbol)  BDEVICE  TERMTYP=device-code

                   {,ALTBUF={3270/328x-alt-buffer-size}}
                   {        {0                       }}

                   {,COPYRSP={INPROG}}
                   {         {NO    }}
                   {         {COMPL }}

                   {,CRLF={YES}}
                   {      {NO }}

                   {,MAXERR={max-I/O-error-retries}}
                   {        {3                    }}

                   {,RLSERSP={NO }}
                   {         {YES}}

                   {,UPINTV={number}}
                   {        {1     }}

                   {,WRTONLY={YES}}
                   {         {NO }}

                   output transmission parameters:

                   {,CHPS={number-chars-per-sec}}
                   {      {0                   }}

                   (,CTCHAR=fe-output-control-chars)

                   (,ENDCHAR=fe/be-message-end-chars)

                   {,IDLES={number}}
                   {       {YES   }}
                   {       {NO    }}
                   {       {0     }}
```

(continued)

		(,LAST={NO }) ({YES}) (,OP1={write-op-code}) ({02 }) (,OP2=alternate-write-op-code) (,OP2IND=op2-signal-code) (,OP2RPL=op2ind-replace-code) (,STCHAR=fe/be-message-start-chars) input transmission parameters: (,AIDGRP=number) (,BACKSP={YES}) ({NO }) (,BSCSGMT={YES}) ({NO }) (,EXTCHR={number-input-control-chars}) ({0 }) (,NUMCOL=number-columns) (,SPEC={YES}) ({NO }) (,SYNCIPQ={YES}) ({NO }) DS40 Teletype Dataspeed 40/1 and 2 terminal specifications: (,DS40BUF={YES}) ({NO }) (,DS40FAS={YES}) ({NO }) (,DS40FDX={YES}) ({NO }) (,DS40LTY={YES}) ({NO })

(continued)

		[,DS40MOD={1}] [{2}] [,DS40MS={1}] [{2}] [{3}] [{0}] [,DS40NL={NO }] [{YES}]

AIDGRP

specifies the number of an AIDGRP macro within the Attention Identification Table (ATTIDTBL). Code as a decimal value, 1 to 255. This parameter is valid for IBM 3270s only.

ALTBUF

specifies, for IBM 3270 system CRTs and printers only, the maximum alternate buffer size permitted for output messages to the terminal. It is inspected when an output message specifies an Erase Write Alternate command (X'7E') at the beginning of the message text. If ALTBUF is not coded, such a message will be rejected. It is also used to determine the maximum buffer size for input messages from local 3270 CRTs for which alternate buffer processing is possible. In this case, one of the BTERMs referencing this BDEVICE must be the first BTERM in the local 3270 group. Code as a decimal number from 960 to 32767. The default is 0 (alternate buffer processing not applicable or not available).

BACKSP

specifies whether or not the backspacing delete feature is invoked for the device. The device's translation table must associate a byte configuration of X'BB' with the keyed character chosen to function as the external deleting character. The default is NO.

BSCSGMT

YES specifies that input blocks are to be treated as separate messages. NO specifies that input blocks are to be concatenated until receipt of an ETX or EOT prior to queuing an Intercomm message. The default is NO. (Valid only if TERMTYP=IBM360/DIAL360 or DIAL2780.)

CHPS

specifies the number of characters printable per second by the device. This parameter is valid only for hard copy buffered devices. Code as a decimal value, 0 to 32767. Required for all Dataspeed 40 terminals. The default is 0.

COPYRSP

specifies whether or not the IBM 3270 terminal requesting a COPY transaction is to be notified of the result of the transaction, via an acknowledgement message, and if so, when, as follows:

Code	Notification
COMPL	Upon completion of COPY (default).
INPROG	During COPY: after READ full buffer is completed and copied message is queued for destination terminal (valid only when destination terminal is on different line or control unit from originating terminal; otherwise treated as COMPL).
NO	None.

The acknowledgement messages issued via COPYRSP do not affect any syntax error message that may be issued by the COPYSS subsystem.

CRLF

indicates whether carriage-return/line-feed control characters or a single new line control character is to be used to advance a line. Code YES for a CR/LF combination. The default is NO, specifying a NL character.

CTCHAR

specifies in EBCDIC the terminal-oriented control characters that are to precede those device-destined messages generated only by the BTAM Front End or the Output Utility. If this parameter is not coded, it is assumed that no control characters need be interpolated. Code in hexadecimal. The combined length of CTCHAR, ENDCHAR and STCHAR may not exceed 40 characters.

IBM 3270 terminals: The specified characters are appended before any message where the first three characters are not as follows:

1. A valid 3270 write command (X'F1', X'F5', or X'7E')

   (For IBM 3270 local terminals, the write command is inspected to determine DECTYPE, but is not written.)

2. A write control character (WCC)

3. A set buffer address (SBA) order (X'11')

DS40BUF

indicates whether the Dataspeed 40 Model 1 or 2 printer is buffered (that is, NL idles are not required). Code YES if buffered. If the terminal is a buffered Receive-Only printer, the maximum output message length may not exceed the buffer size (number of lines in the buffer). The default is NO, that is, NL idles are required. See also the DS40NL parameter.

DS40FAS
indicates whether the terminal has a full or monocase ASCII character set. Code YES if full ASCII. This affects NL idles calculations. Also code YES if the printer has 132 column positions. The default is NO.

DS40FDX
indicates whether the Dataspeed 40 Model 2 terminal is used with a full-duplex data link (modem). In this case, EOT as an input message-ending character indicates disconnect. Code YES if so. The default is NO, indicating that the modem is half-duplex.

DS40LTY
indicates whether the Dataspeed 40 terminal is used on a leased line, not switched. Code YES if leased. The default is NO.

DS40MOD
specifies the model number (1 or 2) of the Dataspeed 40 device. The default is 2.

DS40MS
specifies the number of memory segments in the Dataspeed 40 terminal display buffer: 1, 2 or 3, where each segment is 1920 characters (24 lines by 80 columns). The default is 0, indicating that no CRT-oriented idles are required after ESC (escape) sequences. If the terminal is a Receive-Only printer, code 0.

DS40NL
indicates whether printer-oriented idles are required for the Dataspeed 40 terminal configuration. Code NO if the terminal is only a KD (keyboard-display-CRT) with no attached unbuffered printer. The default is YES, indicating that idles are required. Code YES if there is an attached printer on which messages may be printed (PRINT-ON-LINE Key depressed, or DC2 coded in message), or if the terminal is an unbuffered ROP (Receive-Only printer). See also the DS40BUF parameter.

ENDCHAR
specifies in EBCDIC the terminal-oriented control characters that are to be inserted before any ending line control characters (EOB, ETX and/or EOT) in all messages destined for the device, whether generated by the Front End or received from the Back End. If the specified characters already precede the ending line control characters in the message before it reaches the Front End, ENDCHAR is not inserted. Code as an even number of hexadecimal digits. (See also LAST parameter.) The combined length of CTCHAR, ENDCHAR and STCHAR may not exceed 40 characters.

EXTCHR

specifies the number of control characters the device places at the beginning of each input text. The BTAM Front End automatically strips each input text of these characters. Code as a decimal value, 0 to 255. The default is 0.

IDLES

indicates whether or not idles are requested after message control characters (NL, CR/LF, etc.) for an unbuffered device. YES specifies 15 idles. If more or fewer are desired, code as a decimal value, 1 to 255. This parameter is valid only for hard copy unbuffered devices. Do not code for Dataspeed 40 terminals. The default is 0, indicating that idles are not required. NO may also be used to indicate that idles are not required.

LAST

specifies whether message-ending line control character(s) (EOB, ETX and/or EOT) are to be retained or deleted. YES indicates that existing message-ending characters provided in the message are to be retained. NO indicates that the characters are to be deleted and is applicable to certain hardware-compatible Teletype devices or devices supported through the Generalized Front End, where a specific message-ending character controls the terminal transmission mode (conversational or batch). In this case, use of the ENDCHAR parameter can provide a standard message-ending character for all output messages, and prevents transmission of multiple message-ending characters. YES is the default for all terminals, except Dataspeed 40, for which the default is NO.

MAXERR

is the maximum number of consecutive read or write retries allowed, after an I/O error, before a terminal is put down. Code as a decimal value, 1 to 255. The default is 3. For switched asynchronous (start/stop) devices, MAXERR controls the number of read time-outs allowed before the terminal is disconnected.

NUMCOL

specifies number of columns for WLR checking for an IBM 1030 device. Code as a decimal value from 1 to 11. (Valid only if TERMTYPE=IBM1030.)

OP1

specifies the write operation code (DECTYPE) that is normally to be issued to the device. Code in hexadecimal. This parameter is not valid for graphic devices. For operation codes, refer to the IBM OS/VS BTAM SRL manual for DECTYPE operation type. The default is 02. See also the OP2 parameter.

OP2

specifies an alternate write operation code (instead of that specified via OP1) that is to be used at specified times. If this parameter is coded, the OP2IND and OP2RPL parameters must also be coded. These parameters are only valid for certain terminals. Refer to the BTAM Terminal Support Guide for specific values and applicable terminals. Code in hexadecimal.

OP2IND

specifies the bit configuration of a byte code that signals that the write operation specified by the OP2 parameter is to be used as the DECTYPE, and that the signaling byte code itself is to be replaced by the character supplied by the OP2RPL parameter. The byte code should be immediately after the message header as the first character of an outgoing text. Code in hexadecimal. Required if OP2 is coded.

OP2RPL

supplies the character or byte to be substituted for the signaling byte code designated by the OP2IND parameter. Code in hexadecimal. Required if OP2 is coded.

RLSERSP

indicates whether or not a response is required to input of the RLSE system control command, when no messages are queued for transmission to a particular terminal. Code NO if no response is desired. The default is YES, indicating that an informative response is required.

SPEC

indicates whether or not the special DEVSWTCH indicator is to be set on. Usually indicative of special support for a plug-to-plug compatible device. Code YES to set the indicator. If two BTERM macros (one for CRT, one for printer) are defined for an IBM 3275 terminal configuration, two BDEVICE macros must be defined. Both must have SPEC=YES. The default is NO.

STCHAR

specifies in EBCDIC the line control characters that are to precede all messages destined for the device, whether they are generated by the BTAM Front End or passed from the Back End. If the CTCHAR parameter is coded, Front End messages are transmitted in the following order: STCHAR code, CTCHAR code, text. Back End messages are transmitted in the following order: STCHAR code, text. (Control characters may also be generated by a Back End facility, such as ITEM macro CODE parameter, the DEVICE macro FIRST parameter, the MMU member LOGCHARS, or a preformatted message.) If STCHAR control characters are already in the message received from the Back End, they are not duplicated by the Front End. Code in hexadecimal. The combined length of CTCHAR, ENDCHAR, and STCHAR must not exceed 40 characters.

SYNCIPQ

indicates the sequence of logging and queuing of subsystem input and of subsequent line activity. Code YES for slow response method and high message integrity. The line handler logs and queues the input message first, and then acknowledges the block and continues. This method reduces throughput over the line because of delays caused by logging and queuing, particularly if synchronous logging is used. (If SYNCIPQ=YES and TERMTYP=IBM360, a block abort queuing error facility is provided. (See the BTAM Terminal Support Guide.) The default is NO, specifying a quick response method. The line handler dispatches a thread to log and queue the message, which is asynchronously overlapped with acknowledgement of the last block and subsequent line activity.

TERMTYP

identifies to the BDEVICE macro the kind of terminal under definition. Code the symbol specified in the BTAM Terminal Support Guide for the device being defined.

UPINTV

specifies the number of minutes which should elapse before an attempt is made by the AUTOTPUP facility to automatically turn up a terminal which has been put down with I/O errors. This value is only to be used for terminals for which AUTOUP=YES. The default is 1.

WRTONLY

indicates whether the terminal is a receive-only device (printer, punch, etc.). YES indicates that it is, and that a read is not to be issued to the device. NO (default) indicates it is not. Valid for switched asynchronous (start/stop) devices only.

BLINE -- Define Communication Line (BTAM/TCAM/GFE Front End)

The BLINE macro is used in conjunction with the LINEGRP, BTERM, BDEVICE and POLLIST macros to define the Front End Network Table. The coding sequence of these macros varies with the line type. Refer to the BTAM Terminal Support Guide for specific coding sequences. The BLINE macro defines a single communication line and generates a single DECB (or, for graphics lines, a single GACB) through which is conducted all the I/O operations with its associated terminal(s).

When applicable, device mode constraints are defined for each parameter (leased, switched, asynchronous, bisync, etc.). For specific terminal-related information including required parameters, refer to the BTAM Terminal Support Guide.

The form of the BLINE macro is as follows:

(blank)	BLINE	LGNAME=assoc-linegrp-macro-label, UNIT=terminal-type, [,ERRSTAT={YES}] [{NO }] [,ERRTCNT={n }] [{50}] [,GFEVECT=GFE-macro-label] [,OPTION={BISYNC}] [{GRAPH }] [{BTAM }] [,RDONLY={YES}] [{NO }] [,TRSTBL=translation-table-label] leased line parameters: [,AUTO={YES}] [{NO }] [,BUFL=graphic-dev-buffer-length] [,BUFTM={buffer-empty-interval}] [{300 }]

(continued)

```
[,NOETX={DISCARD}]
        {QUEUE  }

[,NUMAC={num-addressing-chars}]
        {0                    }

(,POLLIST=polling-list-label)

[,POLTM={polling-interval}]
        {300              }

[,WRTM={write-interval}]
       {300           }

[,WRTONLY={YES}]
         {NO }

switched line parameters:

[,ACALL={YES}]
        {NO }

(,ALTCALL=alternate-callist-label)

(,ANSLIST=answer-list-label)

[,ANSWER={YES}]
         {NO }

[,BUFTM={eot-exchng-wait-intvl}]
        {300                  }

(,CALLIST=call-list-label)

(,DIALTRM=num-of-terminals)

(,FIRSTRM=terminal-id)

[,FLUSH={YES}]
        {NO }

(,IDSENT=switched-bisync-id)

[,IDVER={NO }]
        {YES}

[,POLTM={m,s}]
        {2,0}

[,SHORTID={YES}]
          {NO }
```

ACALL

indicates whether or not the switched line is an auto-call (terminals may be called from CPU) line. An auto-call terminal automatically answers a call initiated from the CPU. YES indicates the line is an auto-call line. NO, the default, indicates it is not. For a switched asynchronous line, one or both of the following must be specified: ACALL=YES, ANSWER=YES. For switched bisync, ACALL=YES means terminals may be dialed automatically (in which case the BTERM macro, PHONE parameter, must provide telephone number); NO means either that terminals must be dialed manually (if ACALL=YES in BTERM), or may not be called at all. (Applies only to switched lines.)

ALTCALL

supplies the label of a BTAM DFTRMLST macro to be used as an alternate calling list for the next read conversational operation when the current output message contains a character X in MSGHUSR. May be used to signal an alternate terminal operation such as to turn on a paper tape transmitter. An alternate calling list may be shared by two or more lines using the same transmission code. This parameter is optional and applies to switched asynchronous lines only (teletype devices).

ANSLIST

supplies the label of the BTAM DFTRMLST macro specifying the answering list to be used in a read initial or read connect operation. This parameter is required if ANSWER=YES is specified. The answer list may be shared by two or more lines using the same transmission code. For switched bisync lines, see also the IDSENT parameter. (Applies to switched lines only.)

ANSWER

indicates whether or not the switched line is an auto-answer line. An auto-answer terminal is one which dials the CPU; the CPU answers automatically. YES indicates the line is auto-answer. NO, the default, indicates it is not. For switched asynchronous lines, if ACALL=YES is not specified, ANSWER=YES must be specified. If ANSWER=YES is coded, the ANSLIST parameter must provide the address of the answering list. (Applies to switched lines only.)

AUTO

indicates whether or not auto-poll is to be used on the line. Code YES if auto-poll is to be used. NO, the default, indicates that auto-poll is not to be used. If YES is coded, the POLLIST parameter must also be coded. See also the 'list-type' parameter of the POLLIST macro. If YES is specified, all the lines in the line group must be auto-poll lines. (Applies to leased lines only.)

BUFL

specifies, in bytes, the length of the hardware buffer. This parameter is to be coded only if OPTION=GRAPH is specified. If it is not coded, GAM provides a default value of 960. (BUFL applies only to 2260 Locals.)

BUFTM

For leased lines - specifies in timer units (of 1/300th second), the time to allow for hardware buffers to empty. Code as a decimal value, 1 to 32767. The default is 300.

For switched bisync lines - specifies, in timer units (of 1/300th second), the time to wait (before the next test for an output message) between each of three EOT exchanges, when there is nothing to read from the terminal. This allows time for Back End processing of an input message before disconnect, which is automatic after three EOT exchanges with no intervening I/O activity. Code 0 if no wait is to be done (because an output message may not be queued).

CALLIST

supplies the label of a BTAM DFTRMLST macro specifying the calling list to be used in a read conversational operation to indicate to the terminal operator that an input message may be transmitted. The calling list may be shared by two or more lines using the same transmission code. (Applies only to switched asynchronous lines, for which it is required.)

DIALTRM

specifies the number of dial-up terminals (within the BTERM group associated with this BLINE) serviced by the line. Code as a decimal value, 1 to 255. There is no default. (Applies only to switched lines only, for which it is required.)

ERRSTAT

specifies whether or not error statistics are to be collected for display via the STAT system control command. (Refer to System Control Commands.) YES indicates that statistics are to be collected (applies only to leased lines and dial-up TTY/2740 lines). The default is NO.

ERRTCNT

specifies the line error statistics accumulation threshold if ERRSTAT=YES is coded. Code as a decimal value in the range of 1 to 32767. The default is 50.

FIRSTRM

specifies the Intercomm terminal-ID assigned to the TERM parameter of the first BTERM macro (within the BTERM group associated with this BLINE) where the id-verification is to start at dial-in connect time. (Applies only to switched lines, for which it is required).

FLUSH

specifies whether the output queue for an auto-answer terminal should be flushed when a new connection is established for that terminal on this line. YES requests queued output flushing (via internal FLSH command). NO, the default, specifies that queued output should not be flushed. (Applies only to switched asynchronous lines.)

GFEVECT
supplies the label of the associated GFE macro. This parameter is to be coded only if UNIT=GFE is also specified. This is required for the Generalized Front End Facility and for Extended TCAM support. (See Generalized Front End Facility and the TCAM Terminal Support Guide.)

IDSENT
if specified, replaces the IDSENT specified in the answer list (referenced by the ANSLIST parameter) for calls answered by this line. If it is not equal in length to the ID in the answer list, it will not be used. Specify in hexadecimal up to 17 character pairs using the transmission code of the device, and ending in ACKO. (Applies only to switched bisync lines.)

IDVER
specifies whether terminal ID verification is to be performed after a read-connect operation for an auto-answer terminal. YES, the default, requests terminal ID verification (matched to value coded for associated BTERM macro DILID parameters) and implies that more than one terminal may dial in on the line. NO specifies bypassing of terminal-ID verification and requests association of the line only with the terminal identified by the FIRSTRM parameter (also code DIALTRM=1). (Applies to switched asynchronous lines only.)

LGNAME
specifies the label of the LINEGRP macro instruction with which the BLINE macro instruction is associated. LGNAME is required.

NOETX
specifies whether accumulated segments of an input message are to be queued or discarded, if an EOT is received without an ETX. The default is QUEUE. (Applies only to leased line CPUs.)

NUMAC
specifies the number of addressing characters supplied by each associated BTERM macro, POLL parameter. Code as a decimal value, 0 to 255. The default is 0. (Applies to leased lines only.)

OPTION
indicates the communication line type. Except for synchronous Uniscope 100s, the assigned code of this parameter must be identical to the value of the associated LINEGRP macro, OPTION parameter (BTAM, BISYNC or GRAPH). The default is BTAM. For synchronous Uniscope 100's, the LINEGRP macro OPTION parameter must be BISYNC and this BLINE macro, OPTION parameter, must be BTAM.

POLLIST
supplies the label of the POLLIST macro that describes the polling list for the line. (Applies to leased lines only.)

POLTM

For leased lines - specifies, in timer units (of 1/300th second) the duration permitted to elapse before another poll is initiated, that is, the time interval used after a negative polling pass. This parameter is to be used only when polling is performed and the polling list is an open list. Code as a decimal value, 1 to 32768. If the polling list is a wrap list, code 0. The default is 300.

For switched bisync lines - specifies, in minutes (m), and seconds (s) the maximum time to wait for a call from a remote station. At the expiration of this time interval, the read is cancelled and the line handler will check for output. The default is 2 minutes.

RDONLY

indicates whether or not all terminals on the line are to be treated as input-only devices. YES specifies read-only devices and causes output messages sent to these terminals to be rerouted to the control terminal. The default is NO.

SHORTID

indicates, for Wiltek terminals only, whether the first four characters only of the terminal-ID are to be used for identification purposes, the fifth to be used as a component identification character. The default is NO.

TRSTBL

supplies the label of the beginning of the translation table (input, followed by output or a PMISTOP) to be associated with the terminal or terminals on the line. If OPTION=GRAPH has been specified, this parameter should not be coded.

UNIT

specifies the type of terminal accessed by the line. See the BTAM Terminal Support Guide for coding instructions. This parameter is required.

WRTONLY

except for the IBM 7770 (for which WRTONLY has a special purpose), indicates whether or not all terminals on the line are to be treated as write-only devices. YES indicates the terminals are to be treated as write-only devices. NO, the default, indicates they are not. (This applies to leased lines only.) For the IBM 7770 audio response unit, WRTONLY=YES allows queued messages to be retrieved by telephone.

WRTM

specifies, in timer units (of 1/300th second), the duration of time allowed to elapse between attempted write transmissions when a terminal is busy. This parameter may be coded for any buffered hard copy device. Code as a decimal value, 1 to 32767. The default is 300. (Applies to leased lines only.)

BTERM -- Define Terminal (BTAM/TCAM/GFE Front End)

The BTERM macro is used in conjunction with the LINEGRP, BLINE, BDEVICE and POLLIST macros for the Front End Network Table definitions. The BTERM macro defines a single terminal and provides operational information concerning that terminal.

The coding sequence of these macros varies with the line and device type. For general feature information and specific terminal-related information, refer to the BTAM Terminal Support Guide.

The form of the BTERM macro is as follows:

(blank)	BTERM	General Parameters: TERM=five-character-terminal-id ,DEVIND={relative-'bdevice'-macro-num} {0 } ,QNUM=relative-queue-number (,ALT=alternate-terminal-ID) {,AUTOUP={YES}} { {NO }} {,CONTROL={YES}} { {NO }} {,CONV={YES}} { {NO }} {,CRT={YES}} { {NO }} {,DDQRSRT={BEGIN}} { {LEAVE}} {,EOFMSG={YES}} { {NO }} {,ERRSTAT={YES}} { {NO }} {,HARDCPY={YES}} { {NO }}

(continued)

		(,LOCK=verb) {,SEGLOCK={NO }} { {YES}} {,TPUP={YES}} { {NO }} message restart specifications: {,LOG={NO }} { {YES}} {,LSYNCH={YES}} { {NO }} {,RESTART={NO }} { {IFPOSBL}} { {YES }} multiregion parameters: (,MRPASSW=password) leased line terminal parameters: (,ADDR=addressing-chars) {,POLL={polling-chars}} { {00 }} switched line terminal parameters: ,DILID=caller-id-seq-in-ebcdic {,ACALL={YES}} { {NO }} {,BLK2741={YES}} { {NO }} {,BUFMODE={YES}} { {NO }} {,CINTVL={call-interval}} { {0 }} (,HEXID=called-id-seq-in-ascii) (,IDRCVD=(sw-bi-id-ACKO,sw-bi-id-ACKO,...))

(continued)

		(,IDSENT=switched-bisync-id-ENQ) {,ONETURN={YES}} { {NO }} (,PHONE=telephone-number) (,POLLSEQ=polling-sequence) {,PORTS=(addressing-char,send/receive indication,} { alternate port-id,...)} {,RDFIRST={YES}} { {NO }}

ACALL

Switched asynchronous line: specifies whether or not the terminal is an auto-call (can be called from CPU) device. Code YES for an auto-call device. NO, the default, indicates that it is not an auto-call device. (See also CINTVL, HEXID and PHONE parameters.)

Switched bisync line: specifies whether or not the device can be called from the Intercomm CPU (auto- or manual-dial). NO, the default, indicates that it cannot. (See also CINTVL, IDSENT, IDRCVD, and PHONE parameters.)

ADDR

specifies, in hexadecimal, the addressing characters of the terminal if these characters are different from the polling characters assigned to the POLL parameter. (Leased line terminal only.)

ALT

specifies the ID of the terminal to be used as an alternate terminal if output for the original terminal cannot be transmitted to it. See also the RESTART parameter. The alternate terminal must be of the same device type as the original, as specified via the TERMTYPE parameter of the associated BDEVICE macro. The specification of an alternate terminal may be overridden by the TDWN command, ATD subparameter (See System Control Commands.) Messages routed to an alternate terminal remain on that terminal's queue even though a TPUP may be performed for the original terminal.

AUTOUP

specifies whether or not the terminal is to be automatically reactivated after being turned down due to I/O errors, after the time interval specified in the BDEVICE macro, UPINTV parameter elapses. Code YES to use this facility. NO, the default, is to be specified for switched line terminals, except those for which ACALL=YES.

BLK2741

specifies whether or not the message blocking feature for an IBM 2741 dial-up terminal is to be used. Code YES if the feature is to be used. The default is NO.

BUFMODE

specifies whether or not buffer mode operation is to be used on asynchronous dial-up lines with teletype terminals. If YES is coded, the terminal always operates in buffer mode. The default is NO.

CINTVL

specifies, in seconds, the time interval between successive calls to a switched line terminal. This parameter is valid only if ACALL=YES has been specified. Code as a decimal value, 1 to 27962. The default is 0. A nonzero value is required for switched line terminals except for switched bisync terminals which are only to be called when output is queued, in which case 0 must be specified.

CONTROL

specifies whether or not the terminal is the control terminal. Only one BTERM macro can specify YES. The default is NO.

For the control terminal, the terminal-ID specified via the TERM parameter must be identical both to the code setting of the symbolic variable &CNTL within the SETENV member and to the value assigned to the SPALIST macro CCNID parameter.

CONV

specifies whether or not the terminal is to be considered a conversational terminal. Code YES if conversational processing is desired (see the BTAM Terminal Support Guide). The default is NO.

CRT

Specifies whether the device is to be treated as a CRT. YES indicates that every message output to the terminal must be preceded by an input message. NO specifies that the terminal may receive two output messages with no intervening input message. The default is NO.

DDQRSRT

specifies the action to be taken if a terminal is put down while transmission of output messages on a DDQ is in progress. BEGIN specifies that retransmission from the beginning of the DDQ is desired whenever the terminal is subsequently put up. LEAVE, the default, specifies that transmission will continue with the next message in the DDQ when the terminal is reactivated.

DEVIND

specifies the ordinal position relative to 0 within the series of coded BDEVICE macros and indicates the associated BDEVICE defining the device type of the terminal. For example, DEVIND=9 associates the terminal with the tenth BDEVICE macro. Code as a decimal value, 0 to 255. The default is 0.

DILID

Switched asynchronous line: specifies the identification sequence used to check the validity of a calling terminal (Auto-answer). Translate the sequence into EBCDIC and code in hexadecimal. (Refer to DFTRMLST macro 'ridseq' device dependent operand, IBM OS/VS BTAM SRL manual.) This parameter is required.

Switched bisync line: specifies the identification received when answering a call from this terminal. This parameter is required if ACALL=NO is specified. If ACALL=YES is specified, and the DILID is not coded, the first (or only) ID specified in the IDRCVD parameter is the default. If DILID is coded, it must end in ACKO, even though the ID received when answering ends in ENQ. Code in hexadecimal in the transmission code of the device.

EOFMSG

specifies whether or not a special end of transmission message is to be sent automatically when the remote terminal is locked to a verb and transmits an EOT or disconnects, or an unrecoverable I/O error occurs. The message is automatically queued for the locked verb's subsystem in the format:

verb$*EOT@

where $ is the system separator character and @ is the End of Transmission sequence. The default is NO. YES requests this facility, and in this case, the LOCK parameter must also be coded. (IBM 2780 and CPU-to-CPU only: leased and switched.)

EOFMSG processing may be invoked dynamically (for applicable terminals only) via the EOF parameter of the LOCK command (see System Control Commands). Thus, the EOFMSG can also be used as a signal to the subsystem to issue an UNLK command.

ERRSTAT

specifies whether or not error statistics are to be collected for display via the STAT system control command. Code YES if statistics accumulation desired (applicable only to leased line terminals and dial-up TTY/2740 terminals). Refer to System Control Commands. The default is NO.

HARDCPY

specifies whether or not the terminal is a hard copy (output only) device. Code YES if it is a hardcopy device. The default is NO.

HEXID

specifies the auto-call terminal identification sequence, that is, the identification sequence used to check the validity of a terminal when it is called. Translate the sequence into the terminal transmission code and code in hexadecimal, ending with an End-of-transmission sequence. (Refer to DFTRMLST macro 'ridseq' device dependent operand, IBM OS/VS BTAM SRL manual.) A DFTRMLST containing the PHONE and HEXID parameter values is generated for each auto-call terminal. (Required for switched asynchronous line terminals if ACALL=YES is coded.)

IDRCVD

specifies the ID (or IDs) which can be received when calling this terminal. Code in hexadecimal, using the transmission code of the line. The last two characters of each ID must be ACK0, and the maximum length is eight character pairs. (Required for switched bisync line auto-call terminals if ACALL=YES is specified.)

IDSENT

specifies the switched bisync line ID (or connection control characters) to be transmitted when calling this terminal. Code in hexadecimal using the transmission code of the line, and end with ENQ. Only ENQ may be used if line identification is not desired. This parameter is required for the first BTERM for which ACALL=YES is specified. (The value specified applies until a subsequent BTERM specifies a new IDSENT.) (Switched bisync line terminal only.)

LOCK

indicates the verb to which the terminal is locked. The four-character verb specified is prefixed to each incoming message text. Coding this parameter has the same effect as dynamically issuing a LOCK command, and can be dynamically overridden by a LOCK or an UNLK command issued by a terminal or the Back End. (See System Control Commands.)

LOG

specifies whether or not system log entries are to be made for this terminal. Code NO to bypass logging. The default is YES.

LSYNCH

specifies whether or not log records for this terminal are critical, that is, must be added to the current buffer and written to the system log immediately. Otherwise, the records will accumulate until the buffer is full. YES specifies write immediately. NO, the default, specifies add to the current log buffer.

MRPASSW
corresponds to the password value (one-to-eight alphameric characters) of the P parameter in the MRPASWRD macro used by the Multiregion Support Facility. Its function is to allow access only to the region specified via the corresponding R parameter in the associated MRPASWRD macro. For further information, see Multiregion Support Facility.

ONETURN
specifies whether or not only one line turnaround (no more than one EOT transmitted in either direction) is to be permitted per connection. The default is NO. (Switched bisync line terminal only.)

PHONE
specifies the telephone number assigned to an auto-call terminal. Code in decimal. (Refer to DFTRMLST macro 'dialchars' device-dependent operand in the IBM OS/VS BTAM SRL manual.) (Switched line auto-call terminal only.)

POLL
specifies, in hexadecimal, the polling characters to be used for the terminal. The default is 00, which signifies an absence of polling characters. (Leased line terminal only.)

POLLSEQ
specifies the polling/addressing sequence for 1050 dial-up terminals only. (Refer to DFTRMLST 'xxyy' device-dependent operand in the IBM BTAM OS/VS SRL manual.)

PORTS
identifies, for WILTEK 300 dial-up terminals, every port on the terminal. For each port on the terminal, code three subparameters separated by commas. The first subparameter is the polling or addressing character, the second indicates whether the port is a "send" (input) or "receive" (output) port. Code S for send port, R for receive port. The third subparameter applies only to send ports and indicates the default routing of responses to that terminal input. Code an index digit, starting with 0, to indicate which receive port should get the output.

QNUM
specifies the ordinal position relative to 1 within the BTAMSCTS CSECT of the specific SYCTTBL macro instruction defining the output queue to be utilized by the terminal. It is more efficient if terminals are defined with dedicated queues (that is, with a unique value for QNUM). Dial-up terminals require dedicated queues. Code as a decimal value, 1 to 32767.

RDFIRST

specifies, for auto-call terminals, whether or not, at connect time, a Read of an input message must be issued before any queued output messages are written. NO indicates Write before Read. The default is NO. (Switched bisync line terminal only.)

RESTART

specifies the type of message restart, if any. YES specifies that restart is required. NO specifies no message restart is to be performed. IFPOSBL specifies that restart is to be performed if a message (log code F2, but no corresponding log code F3) is encountered before the system log read-back-point is found while reading the log backwards during restart. (Restart does not affect messages that had successfully completed processing prior to system failure). This parameter must be consistent between a terminal and its alternate; that is, both must have the same setting specified. The default is YES. (See the Operating Reference Manual.)

SEGLOCK

specifies whether or not an automatic LOCK command is to be generated when the first segment is received and an automatic UNLK is to be generated after the final segment is received. This parameter is only valid for terminals whose input can be queued as message segments (for example, Teletype in Buffer mode, or TCAM input segments). The default is YES. If SEGLOCK=NO is specified, all input segments must contain a verb.

TERM

specifies the unique five-character Intercomm terminal-ID to be associated with the terminal. This identification may be coded as a combination of alphabetic characters and decimal digits; however the first position must be alphabetic. Low-order blanks are not recommended. If a VTAM Front End is also defined, this id may not be duplicated for any LUNIT/LCOMP.

TPUP

specifies whether or not the terminal is to be brought on-line at startup time. Code YES if it is to be brought on-line. If NO is specified, a TPUP command must be used to activate the terminal. This is not necessary for auto-answer dial-up, or Extended TCAM support terminals, for which an internal TPUP is generated when the first input is received (see also the TCAM Support Users Guide). The default is NO.

BTVERB -- Define Verb (transaction-id) for Intercomm Front End

The BTVERB macro is used to define information concerning the Back End routing of messages. It is coded in the Front End Verb Table, BTVRBTB, or the user-coded COPY member USRBTVRB. Messages entering Intercomm must contain a transaction identifier, known as a verb. Each BTVERB macro specifies an allowable verb and logically associates it with the appropriate Back End subsystem.

The form of the BTVERB macro is as follows:

[symbol]	BTVERB	VERB=verb, ,SSCH=high-order-sycttbl-entry-code ,SSC=low-order-sycttbl-entry-code [,AIDTRAN={NO }] [{YES}] [,AUTOLOK={YES}] [{NO }] [,CONV={cycle-time}] [{NO }] [,EDIT={YES}] [{BQ }] [{NO }] [,HDR3270={YES}] [{NO }] [,HPRTY={YES}] [{NO }] [,LOCKEXE={YES}] [{NO }] [,RLSE={YES}] [{NO }] [,SECUR={YES}] [{NO }]

AIDTRAN
applies only to 3270-type terminals for which AIDGRP macro translation has been specified for the special function keys. NO specifies that AID translation is not to be processed when the terminal is locked to this verb. YES, the default, indicates that AID specifications are always to be processed.

AUTOLOK
specifies whether the terminal should automatically be locked to this verb before the input message is forwarded to its subsystem. YES specifies that the terminal is automatically locked. (An UNLK is to be issued by either the terminal operator or subsystem when processing completes.) The default is NO.

CONV
indicates whether or not the verb is a conversational identifier. If the terminal issuing this verb is specified as conversational (refer to the BTERM macro CONV parameter), further input is discarded until an output response is transmitted. If the verb is a conversational identifier, specify in timer units (of 1/300th second) the maximum duration for one complete input/output cycle. Code as a decimal value, 300 to 32767*300. The default is NO.

EDIT
indicates whether or not the Edit Utility is expected to perform editing of the message text accompanying the verb. YES indicates that editing is expected to be performed. NO, the default, indicates no editing is to be performed. BQ indicates that the Edit Utility should be called before the message is queued for the subsystem (satellite region), and that an entry for this verb has been coded in the Back End Edit Control Table (PMIVERBS). If YES is coded, a code of X'00' is inserted in the MSGHVMI byte of the associated message header. If NO is specified, X'FF' is inserted. If BQ is specified, only those messages which are not rejected by the Edit Utility due to errors, or omissions of fields defined as required in the Edit Control Table, are queued for the subsystem. If BQ is specified, basic transaction security cannot be used, since this form of security checking is performed on unedited messages prior to passing them to the subsystem or high-level language interface module that would in turn call the Edit Utility. This restriction does not apply if ESS is used for security checking.

HDR3270
applies only to IBM 3270 CRT terminals. YES specifies that an AID and cursor header is to precede the input text data. NO, the default, indicates that the AID and cursor values should be deleted from the input data (not desired by the processing subsystem).

HPRTY
specifies whether or not this verb is high priority. If YES is coded, a 'P' is placed in the MSGHUSR field of the input message header. This code invokes priority queuing for this message (and its descendants) on any queue with the priority queuing feature. The default is NO.

LOCKEXE

specifies how a message containing this verb is to be processed when the terminal from which it originated is locked to another verb. YES indicates that this verb overrides the locked verb for this message only; that is, the locked verb is not prefixed to the message, as it otherwise would be. YES is the default for the system commands LOCK, UNLK, RLSE, and COPY. NO, the default for all other verbs, indicates that this verb is not exempt from locked verb processing, and that the locked verb is to be prefixed to the message as usual. This processing also applies in a Multiregion environment if the terminal is locked to a satellite region; if LOCKEXE=YES is coded, the message will execute only in the control region.

RLSE

determines screen release processing for terminals defined as CRT-type terminals (via BTERM macro CRT=YES). Such terminals require each output message to be preceded by an input message; otherwise, successive output messages immediately overlay each other. RLSE=YES, the default (except for the system command COPY, for which the default is NO), specifies that the next queued output message may be transmitted in response to this verb. RLSE=NO indicates that the next output message is not to be sent until a RLSE=YES verb is entered. If the CONV parameter is coded on both BTVERB and BTERM macros, RLSE=NO should not be coded or a lockout will occur. The RLSE system control command (or a FECMRLSE) may be used to release subsequent output message(s).

SECUR

indicates whether or not the verb is a system security verb. YES specifies that the verb is acceptable only when it is input from the Control Terminal, and overrides any security specifications for other terminals. The default is NO.

SSC

specifies the low-order subsystem code of the Back End queue that is to retain the message. This code must correspond to the SUBC parameter of the SYCTTBL macro that identifies the Subsystem Control Table entry. Code accordingly as a single alphameric character or as a three-digit decimal value. (Default=000.)

SSCH

specifies the high-order subsystem code of the Back End queue that is to retain the message. This code must correspond to the SUBH parameter of the SYCTTBL macro that identifies the Subsystem Control Table entry. Code accordingly as a single alphameric character or as a three-digit decimal value. (Default=000.)

VERB

specifies the four-character alphameric transaction identifier. If short verbs (that is, verbs of fewer than four characters) are to be entered at the terminal, code the verb padded on the right with Xs to make it four characters in length.

CALLIF -- Conditional Call

The CALLIF macro causes control to be passed to a control section at a specified entry point only if that control section has been linkedited into the load module under execution. If the control section has not been linkedited, no branching will be attempted and execution will continue as if no CALL instruction occurred. The CALLIF macro is an Intercomm expansion of the IBM CALL macro.

The form of the CALLIF macro is as follows:

```
[symbol]  CALLIF     parameters
```

The parameters are as follows:

IBM-call-operands	IBM-call-operands provides the operands as they would normally be coded on a standard list form or execute form of the IBM CALL macro instruction.
[,NO=address]	NO specifies the address to be branched to if the entry point of the routine to be called is unresolved. Default is that the branch is to the end of the macro expansion.

CALLOVLY -- Call Transient Region Overlay

The CALLOVLY macro is used to effect a linkage between one routine and another when one routine has been linkedited into an Intercomm transient overlay region. CALLOVLY causes control to be passed to the Intercomm module PMIOVLAY, which determines whether the object routine has been linked into the root segment or into an overlay region. If it has been linked into the root segment, the CALL is immediately forwarded; if it has been linked into a transient overlay region, an IBM SEGLD macro is issued and an Intercomm INTENQ macro instruction enqueues the region. If PMIOVLAY itself has not been linked into the module under execution, a CALLOVLY instruction becomes equivalent to an IBM CALL macro.

The form of the CALLOVLY macro is as follows:

```
[symbol]  CALLOVLY   parameters
```

The parameters are as follows:

IBM-call-operands	IBM-call-operands provides the operands as they would normally be coded on a standard list form or execute form of the IBM CALL macro instruction.
[,LINK=(14)]	LINK=(14) indicates that the entry point address of the transient overlay handler has been loaded into Register 14. Default is that the macro will load Register 14 with the literal V-type constant (V(PMIOVLAY)).
[,NO=address]	NO specifies the address to be branched to if the module called by CALLOVLY is not in the transient overlay, root segment or linked in an overlay region.

CATCH -- Transfer Ownership of Storage Area from Intercomm to an Application

The CATCH macro is issued in a system with the Resource Management Auditing and Purging Facility in use, to take control of an area of storage belonging to Intercomm. Using CATCH ensures that an area will be freed by the Resource Management purge routine if a program-check or time-out occurs in the thread.

The form of the CATCH macro is as follows:

```
[symbol]  CATCH      parameters
```

The parameters are as follows:

```
ADDR={address}
     {(r)    }
     {(1)    }

,LEN={length}
     {(r)   }
     {(0)   }

[,SPAEXT=(r)]
```

ADDR specifies the address of the area to be transferred; this may be any expression that can be substituted into the second operand field of an LA instruction. The address must be doubleword aligned.

LEN specifies the length, in bytes of the area to be transferred; it may be any expression that can be substituted into the second operand field of an LA instruction. The length is rounded up to a multiple of 8.

SPAEXT specifies a register containing the address of the SPA extension. If this operand is coded, the CATCH macro will generate code to load the address of the routine that does the transfer from the SPA Extension. Default is that it loads a literal V-type constant.

CONVERT -- Convert Index to Address. Address to Index

Resident tables which consist of contiguous fixed-length entries are often referenced through the use of indexing. For example, the 10th entry could be referenced by an index of 10 (or 9, if the index is relative to 0), instead of by the actual entry address. The CONVERT macro provides a convenient method of converting the address of an entry from or to its associated index.

The form of the CONVERT macro is as follows:

```
[symbol]  CONVERT    parameters
```

The parameters are as follows:

```
{INDX={(1)                              }}
{     {(r)                              }}
{     {(index-address[,{HALF }])        }}
{     {               [ {BYTE}]         }}
{                                       }
{ADDR={(1)            }                 }
{     {(r)            }                 }
{     {entry-address  }                 }
```

INDX specifies the index-value to be converted to an address. If a register is provided, that is, (1) or (r), the register contains the actual index value. If an index-address is provided, the index value is obtained from that address. The second INDX subparameter determines whether a one-byte (BYTE) or a halfword (HALF) index is obtained. The default is BYTE. If HALF is specified and the CPU is an IBM 360, the index-address must be on a halfword boundary.

ADDR specifies the entry-address to be converted to an index. If not specified in a register, the entry-address specified is the second operand of an LA instruction.

```
,FIRST={(r)                             }
       {(first-ent[,{DIRECT  }])        }
       {            [ {INDIRECT}]       }
```

FIRST specifies the address of the first table entry. This operand is required. If (r) is coded, the register contains the address of the first table entry. If first-entry is specified, it is a symbol for either the actual first table entry (DIRECT) or of its address (INDIRECT); the default is INDIRECT.

```
[,{LEN={(r)  }  }]
[ {    {value}  }]
[ {             }]
[ {,SHIFT=value }]
```

LEN specifies the length of a table entry. This length may be provided in a register, by a self-defining symbol, or an equated symbol. If LEN is an exponential of base 2, the SHIFT should be specified instead of LEN.

SHIFT specifies the number of bits to shift. This is appropriate if the entry length is an exponential of base 2. For example, if the entry length is 32, which is 2^5, then SHIFT=5 should be specified.

LEN and SHIFT are mutually exclusive. If neither is coded, the conversion is assumed to be of an address to or from an offset relative to FIRST value.

```
[,RELTO={n}]
[       {0}]
```

RELTO specifies a non-negative integer to which the index is relative. Example: If the index of the 10th entry is 10, then RELTO=1 should be coded. Most indexes are relative to either 0 or 1, although other values may be coded. The default is 0.

```
[,RESULT={address                 }]
[        {(address[,{HALF}])      }]
[        {          [ {BYTE}]     }]
[        {(r)                     }]
[        {(1)                     }]
```

RESULT specifies where the result of the conversion (the index or the address) should be placed. The default is (1).

If INDX is specified, the only valid specification for RESULT is (1) or (r). That is, the resulting address is returned in a register. If ADDR is specified, either the resulting index-value may be returned in a register (1) or (r), or else a symbolic address may be specified. If a symbolic address is provided, the second subparameter determines whether a one-byte (BYTE) or a halfword (HALF) index is to be stored. BYTE is the default. If HALF is specified and the CPU is an IBM 360, the field must be on a halfword boundary.

DEVICE -- Specify Device Information

The DEVICE macro is used to create the PMIDEVTB table entries that furnish the Back End with generalized device-type information. Refer also to the DVMODIFY macro.

The form of the DEVICE macro is as follows:

(blank)	DEVICE	parameter TYPE={type-code } {devtype-name} ,LEN=line-length [,BUFSIZE={buffer-size}] [{0 }] [,CHAR={CR}] [{NO}] [{NL}] [,CRT={YES}] [{NO }] [,EOB={NO }] [{YES}] [,EOT={NO }] [{YES}] [,FIRST={NO }] [{YES}]

BUFSIZE

specifies the standard size of the physical buffer of the device if the device is a buffered device. Code as a decimal value, 1 to 32767. The default is 0.

CHAR

specifies what control characters are to be placed at the end of each physical line of text, when necessary. CR specifies that a carriage-return character and a line-feed character are placed at the end of each line. NO specifies that no control characters are generated. NL, the default, specifies that a new line character is generated at the end of each line.

CRT

specifies whether the device is a buffered CRT device. If so, code YES. In this case, the Output Utility checks for screen overflow in addition to buffer overflow (based on the number of lines on the CRT). The new line (NL) character will be suppressed if a line is full (does not end in a blank) or is the last line in the CRT buffer. The default is NO (the device is hard-copy or scrolling is permitted on the CRT).

EOB

specifies whether or not an EOB character is to be interpolated into the output character stream as the next-to-last or last character of each output text destined for this type of device. Code NO if it is not to be inserted. The default is YES, specifying that an EOB character is to be inserted.

EOT

specifies whether or not an EOT character is to be interpolated into the output character stream as the last character of each output text destined for this type of device. Code NO if it is not to be inserted. The default is YES, specifying that an EOT character is to be inserted.

FIRST

specifies whether or not the one or two characters designated by the CHAR parameter are to be interpolated into the output character stream as the first one or two characters of each output text destined for this type of device. Code NO if it is not to be done. Refer to the BDEVICE macro STCHAR parameter. This parameter is not applicable if CRT=YES. The default is YES, specifying that an interpolation is to be done.

LEN

specifies, in bytes, the standard line length of the device. Code as a decimal value, 1 to 255.

TYPE

specifies the devtype-name or the user-assigned numeric code that is to logically represent the type of device under definition. devtype-name is the name of the device type (required for MMU processing); see the STATION macro, IOCODE parameter, for permissable devtype-name values. A numeric code may be any value from 0 to F, except that the following codes are predefined as indicated:

- 2--IBM 2780
- 8--IBM 3270 Model 1
- 9--IBM 3270 Model 2 and up

Refer to the STATION macro, IOCODE parameter.

DISPATCH -- Request Multitask Dispatcher Queuing Service

The DISPATCH macro provides the facility for requesting one of several queuing services from Intercomm's Multithreading Dispatcher, as described in the Operating Reference Manual. The type of service requests available are listed as follows:

- A request for a unit of work to be placed on a specific priority execution queue and executed as soon as priority permits
- A request for a unit of work to be placed on a timer queue and executed upon the elapse of a specified duration of time (see also the INTWAIT macro)
- A request for a unit of work to be placed on an event queue and executed upon the completion of a specified event (see also the INTWAIT macro)
- A request to delete a previously queued request
- A request to terminate control and initiate execution of the highest priority unit of work awaiting execution

Three kinds of queues exist: event, timer and execution queues. There are two event queues, one timer queue and four execution queues. All units of work placed on an event or timer queue remain queued until the event transpires or the duration expires. They are then, depending upon assigned priority, transferred to one of the execution queues. The four execution queues correspond to the highest-lowest priority codes of 0, 1, 2, 3 (see also the SYCTTBL macro PRTY parameter).

It is important to understand the use of registers 0 and 1 when using the DISPATCH macro. Following the execution of any DISPATCH macro, which queues a unit of work and which does not specify EXIT, the address of the work unit's Work Queue Element (WQE) is returned to the routine in progress in register 1. This address must be saved in case the work unit is to be subsequently struck from the queue using the CANCEL parameter of the DISPATCH macro. Care must be taken, however, never to CANCEL a work unit that has already been removed from an execution queue and given execution control, that is, dispatched, since the associated WQE is now a free element and may be reused. Register 0 is therefore used to supply the information necessary to prevent this sort of cancellation: when a work unit is dispatched the address of the WQE in which the work unit has been queued is forwarded in general register 0 to the entry point of that work unit and this address may be used in matching any previously saved WQE addresses.

NOTE: An attempt to cancel a WQE already dispatched will result in a thread purge; that is, on OC2 (via ISK) program check occurs.

The forms of the DISPATCH macro are as follows:

[symbol]	DISPATCH	for an execution, event, or timer queue request: {priority/overlay / 'S' / (r)} {,address / (r) / (0)} {,parameter / (r) / (1) / (13)} [,EXIT] [,INTVL={interval / (r)}] [,ECB={event-control-block-addr / (r)},[INTRNL={IPOST / YES / NO}]] [,LINK=(14)] [,SYS={YES / NO}] for a cancellation request: [CANCEL={wqe-address / (r)}] [,EXIT] [,LINK=(14)] for a termination request: {EXIT / LINK=(14)}

address

specifies the address to be given control when the unit of work is ready to be executed. This address is placed in general register 15 at entry to the dispatched routine. This parameter is positional and is required for all DISPATCH requests, except a CANCEL request. If the address is not resolved in the Intercomm linkedit, an OC2 (via ISK) program check cancels the issuing thread.

CANCEL

specifies the address of a Work Queue Element (WQE) in which a work unit is queued, but which is now to be removed from whatever queue it is currently in, since the work unit itself is no longer to be executed. The use of this parameter is valid only if the work unit has not already been dispatched. Refer to the note at the beginning of the macro. (A WQE is a four-fullword unit that is employed by the Dispatcher to manage a single multithreading request.) For a cancellation request, code as the first parameter.

ECB

specifies the address of an Event Control Block. The use of this parameter will automatically place a unit of work on the event queue and the execution of the work unit will be deferred until after the specified Event Control Block has been posted. When more than one work unit awaits posting of the same ECB the event queue entries for those WQEs are treated as a FIFO queue. The Event Control Block must be on a fullword boundary (see INTRNL parameter).

NOTE: If at the time of the issuance of a DISPATCH macro instruction specifying an ECB, the high-order byte of the Event Control Block has bit 1 set to 1 (already posted), the queuing request is not considered a deferred request and the work unit is placed immediately on the appropriate execution queue.

EXIT

specifies that after the DISPATCH request has been processed, control is not to be returned to the routine issuing the DISPATCH macro instruction. The highest priority unit of work waiting on an execution queue will then be given control. Code exactly as shown.

NOTE: If EXIT is not coded, general register 13 must point to a save area. Also, if a Dispatcher entry point is assigned to the LINK parameter, the EXIT parameter has no relevance if coded (see LINK parameter).

INTRNL

is specified only with the ECB operand. NO indicates that the ECB is posted by another operating system task, that is, running under a TCB other than the one issuing the DISPATCH. In this case, the ECB is said to be an "external" ECB. INTRNL=NO should only be used when necessary since it entails the highest overhead.

IPOST or YES indicates that the ECB is to be posted by the same task issuing the DISPATCH. In this case the ECB is said to be an "internal" ECB. The difference between IPOST and YES is as follows:

- IPOST indicates that the mechanism for posting of the ECB is to be the INTPOST macro. This is the recommended procedure for internal ECB's, as it entails the least overhead, but the ECB must be posted via an INTPOST macro; any other posting mechanism, though physically turning on the post bit in the ECB, will not cause the waiting routine to be dispatched.

- YES indicates that the ECB may be posted by any series of instructions which result in setting the post bit (bit 1) to 1 in the ECB, such as a MVI instruction, a IBM POST macro or an Intercomm INTPOST macro. This option exists only for upward compatibility since it is less efficient than specifying IPOST.

To summarize, for external ECBs, INTRNL=NO is required. For internal ECBs, INTRNL=IPOST is most efficient but requires the ECB to be posted via the INTPOST macro. The default is INTRNL=NO.

INTVL=

specifies, in timer units (1 timer unit=1/300th second), the duration of time a work unit is to be deferred. The use of this parameter will automatically place the work unit on the timer queue. Code as a decimal value, 0 to 4095. If register notation is employed, the value range becomes 0 to 8,388,607 (7 hours, 46 minutes, 2 seconds). It is important to note that it is only when the specified duration expires that the work unit is transferred from the timer queue to the appropriate execution queue. The work unit therefore may be delayed beyond the time interval specified because of the execution of higher priority work units.

LINK=(14)

specifies that the appropriate Dispatcher entry point has been preloaded into Register 14. The Dispatcher entry point addresses may be obtained from the SPA Extension. This parameter is required for a module which is eligible for execution in the Link Pack area. The specific entry points are as follows:

IJKDSP	Place a work unit on a priority execution queue and return control, that is, EXIT not implied.
IJKDSPX	Place a work unit on a priority execution queue and terminate control, that is, EXIT implied.
IJKINT	Place a work unit on the timer queue and return control, that is, EXIT not implied.
IJKINTX	Place a work unit on the timer queue and terminate control, that is, EXIT implied.

IJKWAIT Place a work unit on the event queue and return control, that is, EXIT not implied.

IJKWAITX Place a work unit on the event queue and terminate control, that is, EXIT implied.

IJKPOST INTPOST an internal ECB and return control

IJKPOSTX INTPOST an internal ECB and terminate control, that is, EXIT implied.

IJKCNC Cancel a previously queued request and return control, that is, EXIT not implied.

IJKCNCX Cancel a previously queued request and terminate control, that is, EXIT implied.

IJKRETX Terminate control, that is, EXIT implied.

If this parameter is used, it must be coded exactly as shown.

parameter

specifies a fullword value to be passed to "address" when the unit of work is executed. This value is placed in general register 1 at execution entrance time, and is normally used to point to one or more parameters in a parameter list, or a save area. This parameter is positional and is required for all types of DISPATCH requests, except cancel or termination.

priority/overlay
'S'

specifies the address of a one-byte field that contains the dispatching priority and the overlay reference number assigned to the work unit to be queued. This one-byte field is constructed with its left-most bits containing the priority number and its right-most bits containing the overlay reference number. The number of bits assigned to contain each value within this byte is as follows: The first two bits are for priority codes and the other six are for overlay codes. This bit distribution permits four priority levels, 0-3 (highest-lowest), and provides for an overlay reference range of 0 through 62. (A possible code of X'FF' is not permitted.) However, while the priority/overlay byte may be constructed as just specified (and optionally coded as a literal on the macro), a code of 'S' will most often be all that will be required. This code specifies that the work unit being queued is to be assigned the identical priority/overlay codes already assigned to the routine in progress. These codes will have already been assigned via a prior DISPATCH macro instruction or, alternatively, if no assignment has been done, the Dispatcher will supply an initial code value of X'00'. This parameter is positional and required for all DISPATCH requests, except cancel or termination.

SYS=

YES specifies whether or not the WQE created by this DISPATCH will be placed among the system's WQEs. A code of YES places it in the system WQEs, that is, the WQE will not be owned by the current thread and will not be purged upon thread completion. The default is NO.

DVMODIFY -- Modify DEVICE Macro Specifications

The DVMODIFY macro is used for specific terminals to override and/or augment the specifications in the DEVICE macro for a particular terminal type. It may also be used to set a page limit on an infinite row device (such as the TTY or the IBM 2740), by specifing a maximum number of lines (page length) for output message mapping by MMU. DVMODIFY macros must be coded following the PMISTOP after the STATION macros in the Station table (PMISTATB).

The values coded for the DVMODIFY and DEVICE macro operands are merged to form one device type profile, with precedence given to DVMODIFY macro values. For buffered devices, BUFFRSZ overrides the DEVICE macro BUFSIZE parameter, and LINESZ overrides the DEVICE macro LEN parameter.

The form of the DVMODIFY macro is as follows:

symbol	DVMODIFY	{ALTBUF={YES}} { {NO }} (,BUFFRSZ=n) {,DSECT={YES}} { {NO }} {,HARDCPY={YES}} { {NO }} (,LINESZ=n) (,NOLINES=n)

ALTBUF
: specifies, for 3270 CRTs and 328x printers, whether the BUFFRSZ (and LINESZ) specifications are to be used only when the MAPGROUP macro (for MMU) or REPORT macro (for Output Utility formatting) specify Erase Write Alternate (for alternate buffer processing). Code YES if the device has an alternate buffer, NO if not. The default is NO.

BUFFRSZ
: specifies, for buffered devices only, the device buffer size, or, if ALTBUF=YES, the alternate buffer size for a 3270 CRT or 328x printer. This size replaces the buffer size specified in the DEVICE macro. Code as a decimal number from 80 to 32760.

DSECT

causes the generation of a DSECT for a device modifier table entry. Code YES to cause DSECT generation, NO to suppress it in the resulting assembly. The default is NO.

HARDCPY

specifies whether the device is a hard-copy (infinite row) device, or a CRT. Code YES if hard copy, NO if CRT. The default is NO.

LINESZ

specifies the maximum number of characters that are physically possible for a single line of the device. This number overrides the LEN parameter of the DEVICE macro. If ALTBUF=YES is coded, the LINESZ override is used only when alternate buffer processing is required (see also the BUFFRSZ parameter). Code as a decimal number up to 255.

NOLINES

specifies, for infinite row (hard-copy) devices, the maximum number of lines (rows) that may constitute a logical page length for the device for MMU output processing only. If coded, HARDCPY=YES must also be coded. Code as a decimal number up to 255.

symbol

must be coded as a valid assembler language label and correspond to the associated STATION macro IOCODE parameter.

NOTES:

1) Edit Utility processing uses only the CHAR parameter value (DEVICE macro) to determine the end-of-line character(s) for additional positional and/or keyword field separators. For 3270 CRT input, SBA sequences are converted to the system separator by the EDIT3270 subprogram. DVMODIFY parameter overrides are ignored.

2) Output Utility report processing does not use the NOLINES parameter to determine message length, but does use DVMODIFY BUFFRSZ and/or LINESZ overrides. See also the description of the ALTBUF parameter. The physical message size is determined by the BUFFRSZ parameter (or DEVICE macro BUFSIZE, if BUFFRSZ not coded). For infinite row nonbuffered devices, if neither parameter is coded, the physical message size is determined by the number of lines defined in the Report in use for a particular message. For buffered devices, Output will segment the logical message into two or more physical messages of the size determined by the BUFFRSZ (or BUFSIZE) parameter, if necessary.

3) MMU usage of DEVICE and DVMODIFY specifications is extensively documented in Message Mapping Utilities. See the chapter on installation and the appendices on specific device types in that manual for further details.

EXMVE -- Extended Move

The EXMVE macro generates code to move data areas of any length greater than 0. After execution, registers 0, 1, 14 and 15 will have been changed.

The EXMVE macro does not use the S/370 MVCL instruction, as it is generally slower than the MVC and does not allow overlapping moves. All parameters are positional, and all are required, except the last parameter (error-label). Default omissions must be indicated by consecutive commas.

The form of the EXMVE macro is as follows:

```
| (symbol) | EXMVE | (to-addr),(from-addr),{{(F)},length-addr}
|          |       |                       {{H  }            }
|          |       |                       {(A),length       }
|          |       |                       {R,r              }
|          |       |(,{error-label})
|          |       |( {(r)       })
|          |       |( {BYPASS    })
```

to-addr
: specifies the storage address of the area to which data is to be moved. to-addr may be coded either as an expression legal as the second operand of a LA instruction, such as LABEL or 0(0,2), or as a register name enclosed in parentheses, such as (4) or (RTO). If this parameter is omitted, the to-address is assumed to be in register 14.

from-addr
: specifies the storage address of the area from which data is to be moved. from-addr should be coded the same way as to-addr. If this parameter is omitted, the address is assumed to be in register 15.

error-label
: specifies the storage address of the routine to get control if the length is not greater than zero (positive). The address may be specified in a register enclosed in parentheses. BYPASS may be coded, in which case the next sequential instruction after the macro will be executed (no data will be moved). If this parameter is omitted and the length is not positive, an ISK instruction will be executed within the macro expansion to cause a program check (default).

type code (F, H, R, or A) and length-addr/length/r

The type code specifies how the length is determined, as follows:

Type Code	Meaning and Length
F	Fullword. The length is contained in a fullword of storage. The length value must be an address, given as an expression legal as the second operand of an L or LH instruction, such as LABEL or disp(0,6). If the length is contained in a fullword address, F is the default, and the type code may be omitted.
H	Halfword. The length is contained in a halfword of storage. The length value must be an address, as for F.
R	Register. The length is contained in a general register (0, 2-13), which must be given as the next parameter.
A	Absolute. The length is given as a numeric self-defining term (such as 500 or X'FFFF') or an absolute symbol legal as the second operand of a LA instruction. If the length is given as a numeric self-defining term less than 4096, A is default, and may be omitted.

The length, no matter how specified, must be positive.

GENINDEX -- Generate Subsystem Control Table Indices

The GENINDEX macro is used to automatically generate the SCT overlay index and the SCT binary search index. The GENINDEX macro must be placed after the last SYCTTBL macro in the Subsystem Control Table module (INTSCT).

The form of the GENINDEX macro is as follows:

[symbol]	GENINDEX	[OVLYNDX={NO }] [{YES}]

OVLYNDX

specifies whether the Subsystem Control Table overlay index is to be generated. The default is YES. The GENINDEX macro must be used to generate the overlay index except when there are no resident or dynamically loaded SCTs, or multiple indices for the same Overlay A group are desired.

In these cases, OVLYNDX=NO must be coded and the Subsystem Control Table overlay index must be coded by the user. See the Operating Reference Manual.

GENSEC -- Generate Security Table

The GENSEC macro is necessary only when Transaction Security or Sign-On/Sign-Off Security is used. GENSEC generates the terminal-associated PMISECTB table entries. IF the GENSEC macro is not present, the SECVERBS macro and all STATION macro security parameters are ignored. Only one GENSEC macro instruction is required. It must precede all STATION SECVERBS macros assembled with it. Refer to the SPALIST macro, SONOFF and TRANSEC parameters.

The form of the GENSEC macro is as follows:

```
(blank)   GENSEC     parameter
```

The parameter is as follows:

```
[OPER={DISK}]
[     {CORE}]
```

OPER specifies to the macro processor whether the STATION macro operator security code information is to be conditionally generated along with the full expansion of the STATION instruction, or the operator security code information is to be isolated and generated alone. Code DISK if it is to be isolated and generated alone. If DISK is specified, a single PMISECTB table entry is understood to be under generation and only the operator-security-code information supplied by the associated single STATION macro is to be generated. A code of DISK should be specified, for example, whenever creating a member destined for residence within the operator-security-code SEC000 file. The default, CORE, indicates that the operator-code information is to be conditionally generated. If CORE is specified, PMISTATB and PMISECTB table entries are understood to be both under generation, and only the code expansion of operator-security-code information supplied by those STATION macro instructions also specifying an RBN parameter value is to be inhibited.

HEXCON -- Transform Data from Binary to Printable Hex

The HEXCON macro translates a binary data field to printable hexidecimal.

The form of the HEXCON macro is as follows:

(symbol)	HEXCON	i,r,q

i

specifies the register containing the address of the binary field to be transformed.

q

specifies the length of the area addressed by i. Code as a decimal number in the range 1 to 8.

r

specifies the register containing the address of the print area where the result is to be placed. The length of this area must be twice the value specified for q plus 1.

For example:

If the binary value 01A2E4 is to be printed, then the HEXCON macro could be coded:

```
HEXCON  8,9,3
```

where 8 is the register pointing to the binary field, 9 is the register pointing to the print field, and 3 is the length of the binary field to be printed. The result in the print field would be: F0F1C1F2C5F440. In addition, register i is bumped by the length of the binary field, and register r is bumped by the length of the print field. See also the LAYOUT macro.

ICOMFIX -- Specify VS Page Fixing Group

The ICOMFIX macro defines a group of VS pages to be referenced via the INTERCOMM VS Page Fix facility.

The form of the ICOMFIX macro is:

[symbol]	ICOMFIX	ADDR=,PAGES=,ID=

ADDR=
specifies the CSECT name or entry name of the first resident module of the group.

PAGES=
specifies the number of pages in the group, maximum 32.

ID=
specifies a three-character group identifier.

ICOMLINK -- Create Intercomm Linkedit Deck

The ICOMLINK macro is used to generate a linkedit deck tailored to the user's specific Intercomm linkedit requirements. See the Operating Reference Manual. If the Intercomm Link Pack Facility is used, code parameters for all desired features, then either delete the generated INCLUDE statements for those modules included in the Link Pack linkedit, or code an asterisk in column 1 to prevent inclusion of the module in the Intercomm region linkedit. (See also the LPSPA macro.)

Assembly of the ICOMLINK macro produces the linkage editor control statements for each option selected. The parameters of the ICOMLINK macro combine with the settings within the SETENV and SETGLOBE members to produce the linkedit deck. Thus, SETENV and SETGLOBE should be defined for the installation environment prior to assembly of ICOMLINK.

For parameter descriptions listed below, unless otherwise noted, a code of YES generates the linkage editor statements associated with the facility; NO excludes the facility from the linkedit deck.

A bullet (●) indicates parameters related to system features which require additional specifications for implementation. Tables and/or preformatted data sets and/or execution JCL requirements are described in other system documentation.

The form of the ICOMLINK macro is as follows:

(blank)	ICOMLINK	default load library: {LIBR={ddname}} { {SYSLIB}} linkedit overlay region control: {,ASYNCH={YES}} { {NO }} {,MONOVLY={YES}} { {NO }} {,OVLYSTR={YES}} { {NO }} {,TRANS={YES}} { {NO }}

(continued)

dynamic load facilities:

```
[,DYNLINK={NO }]
[         {YES}]

[,DYNLOAD={NO }]
[         {YES}]
```

application programming languages:

```
[,COBOL={NO }]
[       {YES}]

[,PL1={F  }]
[     {OPT}]
[     {NO }]

[,RECOBOL={NO }]
[         {YES}]
```

front end specifications:

```
[,FETABLE={network-table-name}]
[         {BTVRBTB           }]

[,GFE={YES                        }]
[     {(usermod1[...,usermodn,])}]
[     {TCAM                       }]
[     {NO                         }]

[,TEST={YES}]
[      {NO }]

[,VTAM={(lutype[...,lutype,])}]
[      {NO                   }]
```

file handler/file recovery:

```
[,BACKOUT={YES}]
[         {NO }]

[,DISAM={YES}]
[       {NO }]

[,DSCT={IXDFSCT2}]
[      {IXFDSCT3}]
[      {IXFDSCT1}]
```

(continued)

```
[,FILEREC={YES}]
[         {NO }]

[,FILSTAT={YES}]
[         {NO }]
```

data base management system:

```
[,DBASE={({DL1  },{INQ})}]
[       { {TOTAL} {UPD} }]
[       { {IDMS }       }]
[       { {ADA  }       }]
[       { {M204 }       }]
[       { {META }       }]
[       {(NONE,INQ)     }]

[,DBLIBR={ddname}]
```

online utilities:

```
[,MMU={NO }]
[     {YES}]

[,UTILITY={ALL                    }]
[         {NONE                   }]
[         {utility                }]
[         {(utility[,...,utility])}]
```

debugging facilities:

```
[,DEBUG={YES}]
[       {NO }]

[,LOOPTIM={YES}]
[         {NO }]
```

other system facilities:

```
[,CHKRES={YES}]
[        {NO }]

[,DYNPOOL={YES}]
[         {NO }]

[,GPSS={NO }]
[      {YES}]

[,SAM={YES}]
[     {NO }]
```

(continued)

```
[,SECUR={ESS}]
[       {YES}]
[       {NO }]

[,USRSTRT={NO }]
[         {YES}]
```

Other Special Features:

```
[,AUTOGEN={YES}]
[         {NO }]

[,CICSCF={YES}]
[        {NO }]

[,DATAENT={YES}]
[         {NO }]

[,DDQ={YES}]
[     {NO }]

[,MULTREG={CONTROL          }]
[         {SATLITE          }]
[         {({CONTROL},1LOG)}]
[         { {SATLITE}       }]
[         {NO               }]

[,RJE={YES}]
[     {NO }]

[,STORFCH={NO }]
[         {YES}]
```

ASYNCH

specifies the Asynchronous Overlay Loader. This requires the operating system ATTACH and IDENTIFY macros. This parameter is ignored if OVLYSTR=NO is coded. See also SPALIST macro ASYNLDR operand. The default is NO.

●AUTOGEN

specifies the Autogen Special Feature. Code YES to generate the INCLUDE and INSERT cards to linkedit Autogen as an Overlay A Subsystem (if OVLYSTR=NO, it will be resident). The default is NO, which indicates that Autogen is not used or is to be a dynamically loaded subsystem.

●BACKOUT

specifies the Backout-on-the-Fly File Recovery Facility. If BACKOUT=YES is specified, FILEREC=YES is forced. The default is NO.

●CHKRES

specifies the checkpoint and message restart facilities. (See the Operating Reference Manual.) The default is NO, unless the File Recovery feature is specified, in which case CHKRES=YES is forced.

●CICSCF

specifies the CICS Compatibility Special Feature (CP). The default is NO.

COBOL

specifies use of COBOL interface routines. The default is YES. See also RECOBOL Parameter.

●DATAENT

specifies the Data Entry Special Feature. The default is NO.

●DBASE

specifies a Data Base Management System is in use, as follows:

Code	Data Base Management System
DL1	DL/I
TOTAL	TOTAL
IDMS	IDMS
ADA	ADABAS
M204	Model 204
META	Metabase

If any DBMS is specified, the mode of operation must be specified as follows:

Code	Meaning
INQ	Inquiries only. No updates permitted.
UPD	Updates permitted. The Checkpoint/Restart facility must be included in the system (see CHKRES).

The default is (NONE,INQ); that is, no DBMS is in use.

DBLIBR

specifies the ddname of the library or concatenated library group in which the database modules reside. If this parameter is omitted, the ddname is assumed to be the same as coded in the LIBR parameter.

DDQ

specifies the Dynamic Data Queuing Facility. The default is NO.

NOTE: DDQ modules are automatically included if BACKOUT=YES, CICSCF=YES, or MULTREG=CONTROL. They are also included if the DDQ global in SETENV is preset to 1. See Dynamic Data Queuing.

DEBUG

specifies the debugging facility PMIDEBUG. The default is NO.

DISAM

specifies the DISAM file access method facility. See the Operating Reference Manual. The default is NO.

DSCT

specifies the member name of the particular File Handler Data Set Control Table module to be used, based on the number of files to be handled. See the IXFDSCTA macro. Code one of the following:

Module	Number of Files to be Monitored
IXFDSCT2	50-100
IXFDSCT3	100-200
IXFDSCT1	21-50 (default)

●DYNLINK

specifies the Dynamic Linkedit facility (requires DYNLOAD=YES). See the Operating Reference Manual. The default is YES.

●DYNLOAD

specifies the Dynamic Load facility for subsystems and subroutines. See the Operating Reference Manual. The default is YES.

●DYNPOOL

specifies the Dynamically Loaded Intercomm Pools feature, wherein a pools module is loaded at startup. See the Operating reference Manual. The default is NO (include NEWPOOLS in the Intercomm linkedit).

FETABLE

specifies the name of the Front End Network Table (BTAM and/or VTAM) to include in the Intercomm linkedit, if separately assembled from the Front End Verb Table (BTVRBTB) which is automatically included. The default is BTVRBTB which indicates that the Front End Verb and Network Tables have been assembled in one module.

●FILEREC

specifies the File Recovery Special Feature. The default is NO, unless the Backout-on-the-Fly facility is specified (BACKOUT=YES), in which case FILEREC=YES is forced.

●FILSTAT

specifies the File Handler Statistics Report facility. The default is NO.

GFE

specifies the Generalized Front End Special Feature. Code YES for only the GFE interface itself. Code TCAM for the GFE interface and Extended TCAM Support. Code (usermod1,usermod2,...) for the GFE interface and one or more user modules. Requires &GFE global preset to 1 in SETENV and &BTAM preset to 1 in SETGLOBE. The default is NO.

GPSS

specifies the General Purpose Subsystem is required for processing STRT, STOP, FILE, SNAP, ABND, LTRC and TALY verbs. See System Control Commands. The default is YES.

LIBR

specifies the ddname of the library or concatenated library group in which the modules reside. See the Operating Reference Manual. The default is SYSLIB. To ensure correct inclusion of user-coded tables (SPA, SCT, etc.), the load libraries should be in the order MODUSR, then MODLIB, then MODREL.

LOOPTIM

specifies the unbroken core-loop time-out control facility IJKTLOOP. See the Operating Reference Manual. The default is NO.

●MMU

specifies Message Mapping Utilities. All existing MMU device-dependent modules are included. The user should remove those which are not required. Refer to Message Mapping Utilities for further details. The default is YES.

●MONOVLY

specifies the use of subsystems assigned to Overlay Regions B, C, and/or D. The default is NO.

●MULTREG

specifies the Multiregion Facility Special Feature. Code CONTROL for a Control Region linkedit; SATLITE for a Satellite Region linkedit (no Front End). Code (CONTROL,1LOG) for the Control Region Single Log feature; (SATLITE,1LOG) for the Satellite Region Single Log Feature. If coded, the MRSVC and MULTREG globals must be preset in SETGLOBE. The default is NO.

OVLYSTR

specifies the use of an overlay linkedit structure for certain Intercomm and/or user modules. The default is NO (recommended if executing under MVS).

PL1

specifies use of PL/1 services. Code F if the F-level compiler is used; code OPT if the optimizing compiler is used. See the Operating Reference Manual. The default is NO, indicating that PL/1 subsystems are not in use.

RECOBOL

specifies use of reentrant COBOL services (requires COBOL=YES). See the Operating Reference Manual. The default is YES.

●RJE

specifies the RJE Special Feature facility (for OS systems only). The default is NO.

●SAM

specifies the System Accounting and Measurement Facility. See the Operating Reference Manual. The default is NO.

●SECUR

specifies security functions to be used. ESS specifies the Extended Security System; see Extended Security System. YES specifies basic security functions are in use; see the Operating Reference Manual. The default is NO.

●STORFCH

specifies the Store/Fetch Facility. See Store/Fetch Facility. STORFCH=YES is automatic if either AUTOGEN=YES, CICSCF=YES, MMU=YES, or DATAENT=YES is coded. The default is YES.

TEST

specifies what Front End facilities are in use, as follows:

Code	Meaning
YES	Test Mode processing modules or Basic TCAM Support; no Front End (see also MULTREG parameter).
NO	BTAM/GFE/Extended TCAM and/or VTAM Front End (if VTAM=YES) or the BTAM simulator is used.

See the Operating Reference Manual. The default is NO.

TRANS

specifies the use of a transient subroutine overlay region. This parameter is ignored if OVLYSTR=NO has been specified. The default is NO.

USRSTRT

specifies the user exit USRSTART is to be called at system startup. See the Operating Reference Manual. The default is YES.

●UTILITY

specifies use of certain utilities. Code ALL if all utilities in the following list are in use. Code NONE if no utilities are to be used. If selected utilities only are to be used, code in a parameter sublist, as follows:

Code	Meaning
CHANGE	Change Utility
DISPLAY	Display Utility (Change Utility required)
EDIT	Edit Utility
OUTPUT	Output Utility
PAGE	CRT Page Browsing Special Feature.

The default is (EDIT,OUTPUT,CHANGE,DISPLAY). For Intercomm statistics display command processing, OUTPUT is required.

●VTAM

specifies the VTAM Front End. To include it, code a list of all logical unit type codes to be used. Valid logical unit type codes are given in the description of the VTLSB macro LUTYPE parameter in the SNA Terminal Support Guide. If coded, the &VTAM global in SETGLOBE must be preset to 1. The default is NO.

ICOMPOOL -- Define A Set of Fixed-Length Areas of Storage

The ICOMPOOL macro is used, in a system supporting Resource Management user-defined pools, to create a pool of main storage areas that may be acquired by STORAGE macros and released by STORFREE macros. In addition to the DS instructions that reserve space for the pool blocks, this macro generates indexing information and space for accumulating pool-use statistics. Normally several pools of different-sized blocks are defined; ICOMPOOL macros for each pool size must be in the same module, arranged by increasing block size. The maximum number of ICOMPOOL macros that may be coded is 256.

ICOMPOOL macros are coded in a member called NEWPOOLS (if resident in the Intercomm region) or a member called ICPOOLnn, if dynamically loaded at Intercomm startup. See the Operating Reference Manual.

The form of the ICOMPOOL macro is as follows:

(blank)	ICOMPOOL	LEN=block-size, NUMBER=number-of-blocks (,LOWLIM=minimum-allocation)

LEN
: is the length of the blocks in the pool. The blocks are multiples of doublewords; the length will be rounded up if not a multiple of 8. Maximum size is 256K less 8 bytes.

LOWLIM
: specifies the smallest request size that will be filled from this pool. If this operand is not coded, Resource Management will try to fill from this pool any storage request whose length is greater than the LEN parameter of the previous ICOMPOOL (or zero, if this is the first) and no greater than the LEN parameter of this one. A length equal to LOWLIM will not be allocated from the pool; if LEN and LOWLIM have the same value, an error message occurs at assembly time.

NUMBER
: is the number of blocks in the pool.

INTDEQ -- Dequeue

The INTDEQ macro is used in conjunction with the INTENQ macro to signal to the enqueuing-dequeuing module (PMINQDEQ) that a resource, having already been INTENQed (and subsequently accessed), is no longer needed. For every INTENQ macro issued for a resource, there must be an INTDEQ macro subsequently issued for the same resource. If the INTDEQ is successful, register 15 will contain a return code of 0. If, however, a previous INTENQ was not issued or if the program issuing the INTENQ times out, register 15 will contain a return code of 4.

The form of the INTDEQ macro is as follows:

[symbol]	INTDEQ	{resource-id-address} {(r) } {(1) } [,LINK=(15)] [,RLEN={(r) }] [{resource-length}] [{16 }] [,SYSTEM={option}] [{NO }]

resource-id-address
: specifies the address of an identifier denoting the resource upon which an INTENQ macro instruction was previously issued, or a register containing the address.

LINK
: specifies that the DEQUEUE entry point (in PMINQDEQ) has been preloaded in register 15 (may be obtained from SEXDEQ in the SPAEXT).

RLEN
: specifies, in bytes, the length of the resource-id to be dequeued. Code as a general purpose register (r) which contains the length value, or as an equated symbol, or as a decimal value from 1 to 44. The length must be the same as that used for the original INTENQ for the resource-id. The default is 16 (also used for pre-release 9 versions of INTDEQ).

SYSTEM
: specifies the same code as that assigned to the SYSTEM parameter on the associated INTENQ macro instruction. The default is NO.

INTENQ -- Enqueue

The INTENQ macro is used in conjunction with the INTDEQ macro to serialize the use of a particular resource and, if necessary, delimit the number of concurrent users of that resource. The INTENQ macro is a request to be placed upon a resource queue. Control is not returned to the issuer of the INTENQ until all previous requestors for the same resource have been given resource access. However, if the SHARE parameter is coded, all previous requestors may or may not have dequeued themselves by the time control is received. When a requestor is placed on a queue, all registers are saved; register 13 must point to a save area. The INTENQ macro expansion uses registers 0, 1, 14 and 15. No return code is employed, except if the TEST option is used.

The form of the INTENQ macro is as follows:

```
[symbol]   INTENQ   {resource-id-address}
                    {(r)                }
                    {(1)                }

                    [,LINK=(15)]

                    [,RLEN={(r)             }]
                    [      {resource-length}]
                    [      {16              }]

                    [,SHARE={number}]
                    [       {(r)   }]
                    [       {1     }]

                    [,SYSTEM={EXCL}]
                    [        {CHNG}]
                    [        {SHR }]
                    [        {YES }]
                    [        {NO  }]

                    [,TEST={YES}]
                    [      {NO }]

                    [,TIME={max-duration}]
                    [      {(r)         }]
                    [      {(0)         }]
                    [      {NO          }]
                    [      {0           }]
```

resource-id-address
specifies the address of an identifier denoting the resource to be enqueued upon, or a register containing the address. The manner in which the identifier is constructed is not important as long as all users of the resource use the identical bit configuration and length for the identifier.

LINK
specifies that the ENQUEUE entry point (in PMINQDEQ) has been preloaded in register 15 (may be obtained from SEXENQ in the SPAEXT).

RLEN
specifies, in bytes, the length of the resource-id to be enqueued. Code as a general purpose register (r) which contains the length value, or as an equated symbol, or as a decimal value of 1 to 44. The default is 16 (also used for pre-release 9 versions of INTENQ).

SHARE
specifies the maximum permissible number of concurrent users of the resource within the Intercomm region issuing the INTENQ. Two or more subsystems requesting an identical resource may specify different sharecounts: each will be honored accordingly. Code as a nonzero decimal value from 1 to 255. The default is 1, specifying that when execution control is returned to the program issuing the INTENQ macro instruction, it will get exclusive ownership of the resource.

SYSTEM
specifies systemwide control of the resource, that is, whether or not any modules executing in other regions which may be using the identical resource are to be allowed to share the resource or are to be prevented from using the resource, or if no systemwide control is desired. Code YES or EXCL if other regions are not to share the resource; an OS System ENQ macro instruction for exclusive control is issued with the "gname-address" parameter pointing to the eight-character name of 'INTERCOM' and the "rname address" pointing to the resource identifier. If a regionwide SHARE count greater than 1 is specified, the ENQ is issued only for the initial request; an OS DEQ macro instruction is not issued until all modules which specified YES or EXCL when requesting the resource have INTDEQed. If EXCL (or YES) is coded, and the region currently has systemwide shared control, an OS ENQ RET=CHNG is issued to gain systemwide exclusive control. Code SHR if systemwide sharing of the resource is desired (the regionwide SHARE count may be coded as greater than 1); an OS ENQ SHR is issued.

Code CHNG to request a change from systemwide shared control to exclusive control by this region. If CHNG is coded with SHARE=1 (default), PMINQDEQ will wait for all previous owners in this region (if any) to dequeue from the resource before an OS ENQ RET=CHNG is issued. If CHNG is coded with a SHARE count greater than 1, an immediate OS ENQ RET=CHNG is issued. A RET=CHNG request will not take effect until all other regions have dequeued from the resource, and forces systemwide exclusive control for the issuing region, even though sharing of the resource within that region may still be in effect. If CHNG is coded, and the region already has systemwide exclusive control, another ENQ will not be issued; only the SHARE count (requested and/or in effect) governs whether the requestor gains immediate or delayed control of the resource. Note that although coding EXCL, YES or CHNG ultimately results in the issuing region gaining exclusive systemwide control of the resource, the choice of the parameter value indicates whether another program could also issue a shared or exclusive control request (use CHNG) or only multiple threads of this program will issue the request (code EXCL). NO is the default (systemwide control is not desired).

TEST

specifies if the user desires only a return code indicating whether the resource is currently available or is in use. The issuer's request is not actually enqueued. If YES is coded, the SHARE, TIME and SYSTEM parameters need not be specified. The return code will be placed in register 15; a code of zero indicates that no other user is currently using the resource; otherwise a code of 4 is returned. The default is NO, indicating an in-use test is not desired.

TIME

specifies, in seconds, the maximum duration of time the requesting program can be expected to require the use of the resource after it receives control. This parameter is meaningful only if SHARE=1. If the requesting program does not issue an INTDEQ macro instruction within the time period specified, one will be issued automatically (snap 114 enqueue time-out and thread purging occurs). There are four coding options:

1. TIME=0 causes the time-out value to be taken from the SPA field SPANQTIM. Refer to the SPALIST macro, NQTIM parameter. This is the default specification.

2. TIME=max-duration specifies a decimal value 1 to 4095 seconds.

3. TIME=(0) or TIME=(r) specifies register notation: the register may contain a value from 1 to 27,962 (7 hours, 4 minutes, 2 seconds). A negative halfword value causes the enqueue routine logic to bypass time-out processing.

4. TIME=NO suppresses enqueue time-out; the TCTV coded for the associated SYCTTBL macro governs subsystem time-out.

INTPOST -- Post Internal ECB

The INTPOST macro is used to post an ECB awaited via the INTRNL=IPOST option of the INTWAIT and DISPATCH macros. This provides the most efficient synchronization technique for two threads within the same Intercomm region. INTPOST may also be used when INTRNL=YES was specified on the DISPATCH macro. If the object ECB is already posted, then no over-posting will take place. Refer to the description of the DISPATCH macro INTRNL operand for full details on usage and restrictions.

The form of the INTPOST macro is as follows:

[symbol]	INTPOST	ECB={(r) } {event-control-block-address} [,CODE={(r)}] [{xx }] [{40 }] [,EXIT] [,{SPA }={(r)}] [{SPAEXT} {YES}]

CODE

specifies a post code to be placed in the ECB. This may be either a two-digit hexadecimal value or may be placed in a register. When a two-digit hexadecimal value is provided, it must be in the range 40-7F and will replace the high-order byte of the ECB. The low-order three bytes of the ECB will be set to binary zeros. If specified in a register, the high-order byte of the register must be in the range 40-7F; the contents of the register will replace the ECB contents. If the high-order byte of the register is binary zeros, a code of 40 will be generated automatically. If code is omitted, a hexadecimal 40 is the default post code.

ECB

specifies the address of the event control block which is to be posted. The ECB referenced must be the object of a DISPATCH or INTWAIT macro which specifies INTRNL=IPOST or INTRNL=YES. The ECB address may be specified in a register (r), or by any value acceptable as the second operand of an LA instruction. This operand is required.

EXIT

This operand has the same meaning as the corresponding operand of the DISPATCH macro.

SPA
SPAEXT

specifies whether the entry point of the INTPOST routine is to be obtained from the SPA Extension. These operands are mutually exclusive. If SPA=YES or SPAEXT=YES is coded, then the entry point will be obtained from the SPA Extension, and a base register and a prior USING statement must have been established for the SPA or SPA Extension, respectively. If SPA=(r) or SPAEXT=(r) is coded, the entry point will be obtained from the SPA Extension and the designated register must contain the address of the SPA or SPA Extension, respectively (USING statement for that register not required). If neither SPA nor SPAEXT is coded, then a V-type address constant will be generated in-line to obtain the entry point.

INTTIME--Request Time and Date

The INTTIME macro is, for MVS users only, an optimization of the IBM TIME macro (does not use the TIME SVC). Like the standard TIME macro, processing results are returned in registers 0 and 1. A processing CSECT is generated (TIMESECT) which is shared among all modules in the Intercomm linkedit.

The form of the INTTIME macro is as follows:

(symbol)	INTTIME	type(,address) (,BUSY=branch-address)

type
: specifies the form of the time to be returned as in the IBM TIME macro: TU, BIN, DEC, MIC. If this positional parameter is omitted or no parameters are coded, DEC is the default.

address
: specifies an eight-byte (fullword-aligned) area where the time is to be stored if a type of MIC (microseconds) is requested.

BUSY
: specifies a branch address to be taken for reentrant programs if the TIMESECT routine is in use by another program, or by another thread of the same program. This parameter is recommended for reentrant modules executing in a non-MVS environment (where the TIME SVC is used), and for subtasked code executing in all environments on an AP or MP CPU system.

INTWAIT -- Temporarily Relinquish Control

The INTWAIT macro causes the issuing module to temporarily relinquish control until either an ECB is posted or a time interval expires. It assumes only that the caller's register 13 is pointing to a save area.

The form of the INTWAIT macro is as follows:

(symbol)	INTWAIT	{INTVL=time-interval } {ECB=ecb-address(,INTRNL={YES })} {IPOST} {NO } (,LINK=(14)) (,SYS={YES}) ({NO })

The INTWAIT macro is a convenient way of replacing the following often occuring sequence:

```
          STM       14,12,12(13)
          DISPATCH  'S',label,(13),EXIT,{ECB=  }
                                        {INTVL=}
label     LR        13,1
          LM        14,12,12(13)
```

The INTWAIT parameters have the same meaning as the corresponding parameters on a DISPATCH macro, except as noted below.

ECB

specifies the address of an Event Control Block. The use of this parameter will automatically place a unit of work on the event queue and the execution of the work unit will be deferred until after the specified Event Control Block has been posted. Although one or more work units may await posting of the same ECB, there can be no guarantee which work unit will receive initial control. The Event Control Block must be on a fullword boundary.

NOTE: If at the time of the issuance of a DISPATCH macro instruction specifying an ECB the high-order byte of the Event Control Block has either bit 0 or bit 1 set to 1, the queuing request is not considered a deferred request and the work unit is placed immediately on the appropriate execution queue.

INTRNL

See DISPATCH macro, INTRNL operand.

CAUTION: Improper specification of INTRNL=YES can cause a WAIT state, time-outs and other error conditions.

INTVL

specifies, in timer units (one timer unit=1/300th second), the duration of time a work unit is to be deferred. The use of this parameter will automatically place the work unit on the timer queue. Code as a decimal value, 0 to 4095. If register notation is employed, the value range becomes 0 to 8,388,607 (7 hours, 46 minutes, 2 seconds). It is important to note that only when the specified duration expires is the work unit transferred from the timer queue to the appropriate execution queue. The work unit therefore may be delayed beyond the time interval specified because of the execution of higher-priority work units.

LINK=(14)

specifies that the appropriate Dispatcher entry point has been preloaded into Register 14. The specific entry points are as follows:

IJKINTX	Place a work unit on the timer queue and terminate control. (EXIT implied.)
IJKWAITX	Place a work unit on the event queue and terminate control. (EXIT implied.)

If this parameter is used, it must be coded exactly as shown.

SYS

See DISPATCH macro, SYS operand.

IXFDSCTA -- Define Data Set Control Table (DSCT)

The IXFDSCTA macro is used by the File Handler and by system or application programs to construct and to refer to fields within the Data Set Control Table (CSECT IXFDSCTA) in which all processing information for files is maintained.

Expansion of the macro produces a DSECT named DSCT describing the DSCT header and one individual data set entry in the Data Set Control Table, and optionally produces the assembled control section to be linkedited with Intercomm.

The form of the IXFDSCTA macro is as follows:

(blank)	IXFDSCTA	[CSECT={CSECT}] [{DSECT}] [,MAXFILE={number}] [{20 }] [,OPTIONS={hex-digits}] [{00000000 }]

CSECT=CSECT
: is coded to cause the control section to be assembled; if omitted or if CSECT=DSECT is coded, no code is assembled (only the Dsect is generated).

The following parameters are valid only if CSECT=CSECT is specified:

MAXFILE
: specifies the number of data set entries (1-32767) to be assembled into the control table. If more than 200 files are to be processed within an on-line Intercomm region, code an IXFDSCT3 module consisting of this macro with the MAXFILE parameter specifying the desired maximum number of files and code DSCT=IXFDSCT3 on the ICOMLINK macro. The default is 20.

OPTIONS

defines the bit settings for the four-byte special options field in the control table. Up to eight hexadecimal digit combinations may be coded, as follows:.

Option	Code
Do not overlap BISAM (single-thread)	xx 40 xx xx
Allow unit record devices in DSCT	xx xx xx 80
Disable automatic initialization	xx xx xx 01
GET: time-slice option	xx xx 80 xx
Single-thread PS reads	xx 80 xx xx
BDAM: prevent exclusive control	xx 02 xx xx
BDAM: force exclusive control	xx xx 02 xx
BDAM: single-thread non-exclusive READs	xx 20 xx xx
BISAM: prevent exclusive control	xx 04 xx xx
BISAM: force exclusive control	xx xx 04 xx
BISAM: Bypass RE-READ option (exclusive control)	xx 01 xx xx

Note: the previously existing R, G and P parameters are replaced under Intercomm Release 9.0 by the system global &FHSTATS in INTGLOBE and SETGLOBE. The setting of the global defines the number of file access statistics accumulators (2, 3 or 5) to be generated for each Data Set Control Table entry. It is released as 5 (required for implementation of the VSAM Local Shared Resources facility). If coded, the R, G and P parameters are ignored.

LAYOUT -- Format a Printable Character String

The LAYOUT macro generates code to initialize an area with blanks, move in character strings, and convert binary numbers to zoned decimal or external hexadecimal form. A list of field descriptors and a list of corresponding argument identifiers are given, followed by keyword arguments to identify the buffer, the number of bytes to blank, and work areas.

The form of the LAYOUT macro is as follows:

```
[symbol]  LAYOUT     parameters
```

The parameters are as follows:

(field-dscrpt[,...,field-dscrpt])

field descriptor consists of a letter defining the field type followed by a number giving the field length, or a quoted string. The format of a field descriptor is as follows:

```
{{B},num }
{{C}     }
{{E}     }
{{I}     }
{{S}     }
{{X}     }
{'string'}
```

The field types are:

B	--	blank
C	--	character string
E	--	edited decimal (i.e., leading blanks)
I	--	decimal (leading zeros)
S	--	skip
X	--	hexadecimal
'string'	--	a quoted string

The field length (that is, the number following the field type) is the length of the printable string, not the length of the argument. For example, if the argument is a four-byte field to be converted to hexadecimal, the field descriptor will be X8, since it takes two characters to represent a hexadecimal byte; if the argument is a halfword to be converted to decimal, and the value of the halfword is unknown, the field descriptor may be I5 (or E5).

If a print field is too small to contain the hexadecimal equivalent of the contents of a register, the high-order bytes are truncated, rather than the low-order. For example, if the field descriptor is X6, and the corresponding argument is (10), the three low-order bytes of Register 10 are converted to printable hexadecimal form.

[,(argument-id[,...,argument-id])]

argument-id specifies the location of an item to be converted to printable form, if necessary, and moved into the print area: an item can either be in core, in which case the identifier is a symbolic name or a base displacement; or it can be a register, and the identifier is the register number or its symbolic name, in parentheses.

The length of the argument is not given explicitly. For an E or I field, when the corresponding argument is a symbolic name, the symbol's length attribute is used: 2 implies halfword, 4 implies fullword, and anything else implies single character.

(continued)

For an X field, the argument length is half the field length; for a C field length, the argument length is the same as the field length.

The B and S fields and quoted strings do not require corresponding argument identifiers, since they describe a field and specify its contents at the same time. The C, E, I, and X fields require identifiers.

[,AREA=print-area-address]

AREA specifies the location of the print area. If this argument is omitted, Register 1 is assumed to contain the area address.

[,BLANK=number-to-blank]

BLANK specifies the number of bytes, starting at the first byte of the print area, to be initialized to blanks. This is done before any of the fields of the print area are filled in.

[,TRANS=translate-table-address]

TRANS specifies the address of a translate table for converting an unpacked field to printable hexadecimal digits, consisting of a constant defined as follows:

```
DC  C'0123456789ABCDEF'
```

If this argument is omitted but required, LAYOUT generates a translate table. When TRANS is omitted, the LAYOUT macro must not be coded in the first 256 bytes of a CSECT.

[,WORK=work-area-address]

WORK specifies the address of twelve-byte doubleword-aligned work area for binary-decimal conversion, etc. If this argument is omitted and a work area is necessary, LAYOUT generates one. To preserve reentrancy, a linkpack module should supply its own area.

LINEGRP -- Define Communication Line Group

The LINEGRP macro is used in conjunction with the BLINE, BTERM, BDEVICE and POLLIST macros to define the Front End Network Table. The LINEGRP macro defines a single teleprocessing communications line group, and generates a single DCB through which all transmission over the associated lines is controlled.

The DDNAME, BUFNO, BUFL, BFTEK, CODE, MODE, LERB, READYQ and EROPT parameters (discussed below) are substituted directly into the DCB. For further information about these parameters, refer to the discussion of the DCB macro in the IBM OS/VS BTAM manual.

A communications line group is defined as the logical association of one or more communication lines. The definition of a line group is constrained by transmission mode and by terminal type. All lines in the group must either be binary synchronous or start-stop, and, if bisync, they must use the same transmission code. All terminals associated with the lines in the line group must use the same line protocol.

The form of the LINEGRP macro is as follows:

symbol	LINEGRP	DDNAME=line-group-ddname ,NUMLN=number-of-assoc-BLINE-instructions ,UNIT=unit-category [,BUFNO=number-of-buffers,BUFL=length-of-buffers] {,BFTEK={buffer-code}} { {D} } {,CODE={transmission-code}} { {EBCDIC} } {,EROPT={error-options-code}} { {E} } [,LERB=LERB-macro-label] {,MODE={(dcb-macro-values)}} { {(IBC,CNTRL,A,A)} } {,OPTION={BISYNC}} { {GRAPH} } { {BTAM} } {,READYQ={routine-address}} { {NONE} } { {0} }

symbol

supplies a label by which the group of lines can be identified. This identification is required. Since other labels are generated from it, this label must not exceed four characters in length. (Refer to the BLINE macro, LGNAME parameter, and the POLLIST macro, LINE parameter.)

BUFL

specifies, in bytes, the length of the buffers in the buffer pool. Do not code if UNIT is 2260 (local), WILT, BKRO, 327L, 373S, 7770, or 2741; otherwise, this parameter is required. Code as a decimal value 4 to 32760. If the value specified is not a multiple of 4, it is rounded up to the next higher multiple of 4.

For async (start/stop) devices, the buffer length should be no greater than the maximum input message length, including message control characters, plus four. Under MVS (which does not permit dynamic buffering), the buffer length should be equal to this value. For non-MVS systems, BUFL may be given as the average input message length (including message control characters), plus four.

For bisync devices, the buffer length must be that of the line (transmission) buffer length, plus eight. (Note that the line buffer length may be different from the I/O buffer length: for a remote IBM 3278 CRT Model 2, the I/O buffer is 1920, but the line buffer is 256; BUFL must be 264).

BUFNO

specifies the number of buffers to be built in the buffer pool assigned to the line group. Do not code if UNIT is 2260 (local), WILT, BKRO, 327L, 373S, 7770, or 2741; otherwise, this parameter is required. Code as a decimal value, 1 to 255.

For async devices (except under MVS), the number of buffers should be the number of associated BLINEs times the number of buffers necessary to hold the longest possible input message. (For example, if BUFL=84, and input messages may be up to 216 bytes long, the number of buffers necessary to hold the longest possible message is 3. BUFNO should therefore be 3 times the number of BLINEs in the line group.) Under MVS, BUFNO should equal the number of associated BLINEs, plus one.

For bisync devices (except under MVS), BUFNO should be twice the number of BLINEs in the line group times the number of buffers necessary to hold the longest possible input message. Under MVS, BUFNO should equal the number of associated BLINEs, plus one.

BFTEK

specifies the buffering technique code. For MVS, dynamic buffering is automatically overridden (not supported by IBM). For MVS under VM, this parameter should be coded as BFTEK= (no operand); it cannot be the last parameter. The list of permissable codes may be found in the IBM OS/VS BTAM manual. The default is D (dynamic buffering requested).

CODE

specifies the transmission code for a binary synchronous line group. This parameter is therefore valid only if OPTION=BISYNC has been specified. The list of permissible codes may be found in the IBM OS/VS BTAM manual. The default is EBCDIC.

DDNAME

identifies the ddname assigned to the DD statement (or two or more concatenated DD statements) that supply the one (or more) unit addresses associated with the one (or more) lines (or local terminal group) that constitute the line group. This parameter is required.

EROPT

specifies the error recovery error recording, and on-line test options to be provided for the line group. This list of permissible codes may be found in IBM's OS/VS BTAM manual. The default is E. If the BUFNO and BUFL parameters have not been specified, user coding of EROPT is ignored and EROPT is set to N. If OPTION=GRAPH this parameter is not valid.

LERB

specifies the address of a LERB macro. The function and coding of the LERB macro may be found in IBM's OS/VS BTAM manual. If this operand is coded, EROPT=C must also be coded for remotes; EROPT=T for locals.

MODE

specifies the transmission mode characteristics for a binary synchronous line group. This parameter is therefore valid only if OPTION=BISYNC. The list of permissible codes may be found in the IBM OS/VS BTAM manual. The default is (IBC,CNTRL,A,A).

NUMLN

specifies the number of BLINE macro instructions (lines) which reference this LINEGRP. Code as a decimal value, 1 to 999. This parameter is required.

OPTION

specifies the type of DCB to be generated for the line group, as follows:

BTAM -- for use with remote asynchronous (start-stop) and with local 3270 devices.

BISYNC -- for use with binary synchronous devices.

GRAPH -- for use with graphic devices.

The default is BTAM.

READYQ

(IBM 3270 Local only) specifies 0 or the address of a user-written routine to receive control when a local 3270 is powered on, for example, to generate an internal TPUP when the interrupt is presented. READYQ=NONE must be specified for those systems in which the READYQ keyword does not exist. The default code is 0 (an operator generated TPUP is required). Refer to the IBM OS/VS BTAM manual for further information.

UNIT

specifies the type of terminal connected to the one or more lines of the group. Code the appropriate symbol as specified in the BTAM Terminal Support Guide, LINEGRP macro device support parameter requirements.

LINKAGE -- Provide Standard Linkage

The LINKAGE macro generates the standard linkage code required by all Assembler Language programs interfacing with Intercomm. It can also perform a number of other functions:

- provide a set of register equates
- issue various USING statements
- set up one or two base registers
- set up specified registers with certain addresses
- obtain core
- zero the core obtained
- provide the PARMLIST, SPALIST, SCTLIST, MSGHEAD and R13 DSECTS

The form of the LINKAGE macro is as follows:

```
[symbol] LINKAGE    parameters
```

symbol supplies an entry point dump-identifier. If coded, it is generated as a character constant. If none is coded, and there is no current CSECT-name, a four-digit decimal number is suffixed to the characters 'EPID' to supply a constant.

The parameters are as follows:

```
BASE={usng-oprd                          }
     {(usng-oprd[,...,usng-oprd])        }

,PARM=(r)

,SPA=(r)

[,DSECTS={sp-sym                         }]
[        {(sp-sym[,...,sp-sym])          }]
[        {NONE                           }]
[        {ALL                            }]
[        {(MSG,R13)                      }]
```

BASE supplies, via a sublist, one or more USING statement base register operands. The following assembler instruction:

USING *,registers

where "registers" are those subparameters supplied, is the first instruction generated by the LINKAGE macro. At least one sublist element must be provided. If multiple registers are designated, the length of the coded text (excluding parenthesis) must not exceed 80 characters. The first base register is set up by loading it with the entry point address taken from register 15. If a second base register is supplied, it will be set to this address + 4096. No other base registers are initialized.

PARM designates the general register (2-12) that is to contain the address of the parameter list passed to the entry point. Refer to the DSECTS parameter, suppression symbol PRM.

SPA designates the general register (2-12) that is to contain the address of Intercomm's SPA. Refer to the TEST parameters. Refer also to DSECTS parameter, suppression symbol SPA.

DSECTS specifies which of the DSECTs noted below are to be suppressed. The LINKAGE macro produces five DSECTs. However, if any of the following five suppression symbols are included with a DSECTS sublist, the associated DSECT is not generated.

For each DSECT not suppressed, an appropriate USING instruction is generated, giving the register specified in the corresponding parameter (except for R13DSECT, for which the register is always 13):

USING DSECT,r

PRM -- supresses the program parameter list DSECT PARMLIST. To generate this DSECT, omit the symbol from the sublist and code the PARM parameter. If DSECTS=PRM, TEST must specify YES.

SPA -- suppresses the SPALIST DSECT. To generate this DSECT, omit the symbol from the sublist and code the SPA parameter.

(continued)

SCT — suppresses the system control table entry DSECT SCTLIST. To generate this DSECT, omit the symbol from the sublist and code the SCT parameter.

MSG — suppresses the message header dscect MSGHEAD. To generate this DSECT, omit the symbol from the sublist and code the MSG parameter.

R13 — supresses the program save area DSECT R13DSECT, To generate this DSECT, omit the symbol from the sublist.

NONE - specifies that no DSECTs are to be suppressed. The code NO is obsolete but remains acceptable.

ALL specifies that all DSECTs are to be suppressed. This code is equivalent to the sublist (PRM,SPA,SCT,MSG,R13).

The default is (MSG,R13).

[,DYN={NO }]
[{YES}]

DYN specifies whether or not dynamic core is to be acquired. Code YES for core to be acquired. Code NO for core not to be acquired. If YES is coded, a conditional GETMAIN is issued for the amount specified by the LEN parameter. The first 72 bytes of this area are used as a save area. Register 13, after LINKAGE, can be expected to contain the address of this area. If NO is coded, it is conventionally assumed that Register 10 points to an area of core that can be similarly used. After LINKAGE, Register 13 can beexpected to contain the area's address. However, if at any LINKAGE point along a succession of LINKAGE's Register 10 equals Register 13, the saving chain is destroyed. It is therefore recommended that no program's save area address be forwarded to be used as another's. Prior to the Subsystem Controller callining a subsystem, Registers (2-12) are zeroed; therefore, nonreentrant subsystems must load Register 10 with an in-line save area address prior to issuing the LINKAGE macro, or DYN=YES must be coded. The default is YES. Refer to the DSECTS parameter, supression symbol R13.

[,GEN={NO }]
[{YES}]

GEN specifies whether or not the PMILINK CSECT is to be generated (valid only for the first LINKAGE macro in a program). Code NO to suppress. NO should be used for dynamically loaded subsystems or subroutines in order to save main storage. The default is YES, specifying that the PMILINK CSECT is to be generated.

[,GPREQ={REGA}]
[{REGS}]

GPREQ designates which of two possible sets of general register equates is to be included in the macro expansion. Code REGA for the REGA set, REGS for the REGS set. There is no default set. Refer to the REGA and REGS macros.

[,LEN=length]

LEN provides the length, in bytes, of main storage to be acquired for initial or subsequent program use. If DYN=NO is coded, this parameter is meaningless and, therefore, not required. It must be coded at all other times. A minimum of 72 is required; however, a minimum of 88 is recommmended in order to provide 72 bytes for a standard program save area and 16 additional bytes for an available four-word area as would be required by the STORAGE macro. Length must be coded as either a decimal number or an equated symbol, from 72 to 4095. The value is rounded up to a multiple of 8 for alignment.

[,MSG=(r)]

MSG designates the general register (2-12) that is to contain the value loaded from the first fullword of the parameter list passed to the entry point. This value will usually be the address of the message that is to be received by the program. Refer to the DSECTS parameter, suppression symbol MSG.

[,SCT=(r)]

SCT designates the general register (2-12) that is to contain the address of the subsystem's associated System Control Table

(continued)

```
[,SPAEXT=(r)]

[,TEST={YES}]
[      {NO }]

[,ZERO={NO }]
[      {YES}]
```

entry. Since this address is loaded from the third full word of the parameter list passed to the entry point, the use of this parameter may be considered valid only if control is received directly or indirectly from the System Controller. If control is received indirectly, then care must be taken to forward the entry address to lower level programs. Refer to the DSECTS parameter, suppression symbol SCT.

SPAEXT designates the general register (2-12) for referencing the SPAEXT DSECT and generates the following USING statement:

```
USING SPAEXT,r
```

TEST indicates whether the address of the System Parameter Area is to be located at the second fullword of the parameter list passed to the entry point or whether it is to be provided via a V-type constant. Code YES for a V-type constant provision. The default is NO, specifying extraction from the parameter list. If DSECTS=PRM, TEST must be specified YES.

ZERO specifies whether or not the core obtained is to be zeroed. If DYN=NO is coded, this parameter is not meaningful. Code NO if the core is not to be zeroed. The default is YES, specifying that the core is to be zeroed.

LOGVERB -- Generate Log Verb Table (LOGVRBTB)

The LOGVERB macro is used to generate the Log Verb Table in order to define subsystem, verb and VMI correspondence to the off-line LOGANAL Utility when edit-before-queuing is requested for the defined verbs. See the Operating Reference Manual for further details.

The form of the LOGVERB macro is as follows:

(blank)	LOGVERB	SUBSYS=subsystem-code, VMI=message-identifier, VERB=transaction-ID

SUBSYS
: specifies the subsystem using edit-before-queing. Code in the format of a 'DC' assembler instruction; for example, C'AB' or X'0134'.

VERB
: specifies the original verb before editing. Code as 1 to 4 characters in apostrophes; for example, 'VRBX'.

VMI
: specifies the VMI inserted in the message header by the Edit Utility. Code in the format of a 'DC' assembler instruction; for example, X'01'.

LPENTRY--Add User Entry Point to LPSPA

The LPENTRY macro instruction defines an entry point of a user module which will be present in the Intercomm Link Pack Area, and is used in conjunction with the LPSPA macro (see Operating Reference Manual).

The form of the LPENTRY macro is as follows:

(symbol)	LPENTRY	entry,{SCT}(,suffix) {SPA} (,SSC={nnn}) {a } (,SSCH={a }) {nnn} {000}

entry
provides the entry point name of the user module. This operand is required.

SCT
indicates that the specified entry point is for a subsystem, and therefore the SYCTTBL macro for the subsystem will generate the VCON for this entry point.

SPA
indicates that the assigned VCON for this entry point is in the USERSPA in the SPA CSECT. (Default)

SSC
defines the low-order byte of the subsystem code when SCT has been specified, and in that case is required.

nnn is a three-digit decimal number in the range 000 to 255

a is an alphanumeric character

SSCH
defines the high-order byte of the subsystem code and is valid only when SCT has been specified. Permissible values are the same as those for SSC. The default is 000.

suffix
provides a suffix to be appended to the characters SPA when SPA has been coded as the second operand. This third operand is optional and is invalid when SCT is coded. The composite SPAsuffix is the label in the User SPA of a DC specifying V(entry). If this operand is omitted, a default suffix will be formed as follows:

1. If entry is five characters or less, then suffix=entry.

2. If entry is six, seven, or eight characters, then suffix is formed by concatenating the first three and the last two characters of entry.

LPINTFC--Define Link Pack Interface CSECT

The LPINTFC macro generates the LPINTFC CSECT containing Intercomm pseudo-entry points for all Intercomm modules eligible for the Link Pack Area, whether or not they actually reside there. The actual physical makeup of the Intercomm Link Pack module is determined by the Intercomm components included in the Link Pack linkedit. (See also the LPSPA macro and the Operating Reference Manual).

The form of the LPINTFC macro is as follows:

[symbol]	LPINTFC	MODS=(mod[,...,mod])

MODS

specifies the Intercomm components for which entry points are to be generated in the LPINTFC CSECT. If none are specified, only the LPINTFC CSECT statement itself is generated; LPVCON macros may be used to add specific user-coded entries into LPINTFC.

The permissable values are as follows:

MSGCOL
RTRVER
FILEHND
FILHNDQI
EDIT
OUTPUT
CHANGE
DISPLAY
CHGDIS
NQDEQ
CONVERSE
SFETCH
DDQ
MMU
PMIEXTRM

See the LPSPA macro for further details. CHANGE, DISPLAY and CHGDIS need not be specified. OUTPUT must be specified if PMIVMI56 is included in the Link Pack linkedit.

NOTE: INCLUDE statements for the LPINTFC Csect module (and LPSTART) should be placed in the Intercomm linkedit after all INCLUDE statements for Intercomm modules (but before OVERLAY statements, if used).

LPSPA -- Define Link Pack System Parameter Area

The LPSPA macro creates the LPSPA CSECT, defining the contents of the Link Pack Area load module used by Intercomm regions and is physically included as a part of that module. See the Operating Reference Manual.

The form of the LPSPA macro is as follows:

[symbol]	LPSPA	[A=A,] [MODS=(mod[,...,mod])]

A=A

specifies whether a CSECT or a DSECT is to be generated. If coded as shown, an LPSPA CSECT statement is generated. Otherwise, an LPSPA DSECT is generated.

MODS

provides a symbolic list of Intercomm modules to be included in the Link Pack Area; if not coded, and A=A is coded, only an LPSPA CSECT statement is generated. To describe the corresponding Intercomm members, which must be included in the Link Pack Area at linkedit time, code one (or a sublist) from the following values:

mod	Intercomm members	Function
MSGCOL	BLMSGCOL	Message Collection
RTRVER	PMIRETRV	Retriever
FILEHND	IXFHND01 IXFB37 IXFLOG IXFVSCRS	File Handler (IXFQISAM not in Intercomm linkedit)
FILHNDQI	IXFHND01 IXFB37 IXFLOG IXFVSCRS	File Handler (IXFQISAM used in Intercomm load module)

(continued)

mod	Intercomm members	Function
EDIT	PMIEDIT PMIFIXED EDIT3270	Edit Utility
OUTPUT	PMIOUTPT PMIVMI56	Output Utility
CHANGE or DISPLAY or CHGDIS	CHANGE DISPLAY FORMAT CRUNCH	Change and Display Utilities
NQDEQ	PMINQDEQ	Enqueue/Dequeue Functions
CONVERSE	CONVERSE	Conversational Support
SFETCH	INTSTORF	Store/Fetch
DDQ	DDQMOD	Dynamic Data Queuing
MMU	MAPIN,MAPOUT, MMUDDM,MMUDDMM, MMUDDMF,MMUDDMT, MMUDDMU,MMUDDMX, MMUED001,MMUED002, MMUED003,MMUED008, MMUTRTS,LOGCHARS	Message Mapping Utilities
PMIEXTRM	PMIEXTRM	Terminal Lookup

For each coded value, the corresponding members in the Intercomm load libraries (MODLIB,MODREL,MODUSR) must be included in the Intercomm Link Pack Area linkedit.

LPSPA may also contain user-defined entries. See the LPENTRY macro.

LPVCON--Define Link-Pack Pseudo VCON

The LPVCON macro instruction generates a user-defined pseudo-entry point in the LPINTFC CSECT (generated by the LPINTFC macro). See also the Operating Reference Manual.

The form of the LPVCON macro is as follows:

(symbol)	LPVCON	entry,SPA(,suffix)

entry

specifies the entry point name of a user module which can be resident in the Intercomm Link Pack Area. This operand is required.

SPA

specifies that a field in the USERSPA in the System Parameter List (SPA CSECT) is reserved for a VCON for the assigned entry point.

suffix

specifies a one- to five-byte suffix to be appended to the characters SPA. The composite SPAsuffix is the label in the User SPA of a DC specifying V(entry). This operand is optional.

If this parameter is omitted, a default suffix will be formed as follows:

1. If entry is five characters or less, then suffix = entry.

2. If entry is six, seven, or eight characters, the first three characters will be concatenated with the last two to form the suffix.

MAPACCT -- Specify System Accounting and Measurement Resources

The MAPACCT macro is used to specify categories of resource usage for the System Accounting and Measurement facility.

The form of the MAPACCT macro is as follows:

[symbol]	MAPACCT	('b1',r,r,...), ('b2',r,r,...),....,('bn',r,r,...)

Each group of parameters within parentheses defines values to be accumulated into a single accounting group or "bucket." The value coded for 'b' must be a character string of one to ten characters and represents the title to be used for that bucket in the final report. Each value coded for the r parameter must be a keyword from the following list and represents a category of resource usage to be included in that bucket. Any number of buckets may be specified but no resource usage category may appear in more than one bucket. The system resource usage categories and their keywords are as follows:

Keyword	Resource--Usage Type
CPUTIME	Total thread CPU time in units of 1/1000 second
HIGHSTOR	Thread high-water mark of core usage. If specified, the STORAGES keyboard must also be specified.
STORAGES	Total number of storage requests
MESSAGES	Total number of messages generated by the thread
PFAULTS	Total number of page faults (VS only)
OLOADS	Total overlay-loads through use of CALLOVLY
LOADS	Total module-loads via the PMIDLOAD module
ENQS	Total ENQS through use of the INTENQ macro (routine)
OPENS	Total of File-OPENs
CLOSES	Total of File-CLOSEs
SETLS	Total QISAM SETLs
QISAMG	Total QISAM GETs

Keyword	Resource--Usage Type
QISAMP	Total QISAM PUTs
BISAMR	Total BISAM READs
BISAMW	Total BISAM WRITE-Updates
BISAMWKN	Total BISAM WRITE-Adds
BDAMR	Total BDAM READs
BDAMW	Total BDAM WRITEs
BSAMR	Total BSAM READs
BSAMW	Total BSAM WRITEs
QSAMG	Total QSAM GETs
QSAMP	Total QSAM PUTs
VSAMG	Total VSAM GETs
VSAMP	Total VSAM PUTs
VSAMPT	Total VSAM POINTs
VSAME	Total VSAM ERASEs
SELECTS	Total File-SELECTs
RELEASES	Total File-RELEASEs
ALLOCS	Total calls to ALLOCATE
ACCESSES	Total calls to ACCESS
FETCORE	Total FETCHs from core
FETDISK	Total FETCHs from disk
STORCORE	Total STOREs to core
STORDISK	Total STOREs to disk
STORUPD	Total STORE UPDATES with length change
UNSTCORE	Total UNSTOREs in core (includes disk UNSTOREs)
UNSTDISK	Total UNSTOREs on disk

Keyword	Resource--Usage Type
MAPINS	Total calls to MAPIN
MAPOTS	Total calls to MAPOUT
MAPENS	Total calls to MAPEND
MAPPRS	Total calls to MAPURGE
MAPCLS	Total calls to MAPCLR
MAPFRS	Total calls to MAPFREE
MPPAGES	Total pages created via MMU
QBLDS	Total number of QBUILDs
QOPNS	Total number of QOPENs
QRDS	Total number of QREADs
QRDXS	Total number of QREADXs (for update)
QWRS	Total number of QWRITEs
QWRXS	Total number of QWRITEXs (for update)
QCLSS	Total number of QCLOSEs
FESCLS	Total calls to FESEND
FEOTPUT	Total calls to PMIOTPUT

The following keyword categories are for user accounting requested via execution of the USRTRACK macro:

Keyword	Resource--Usage Type
USRBKnn	User-defined accumulator (bucket)
USRFNnn	User-defined function (USER exit routine)

where nn is a number in the range of 01 to 10, which relates the keyword to the corresponding USRTRACK macro (See also the Operating Reference Manual.)

MODCNTRL -- Control the Use of Dynamically Loaded Subroutines

This macro requests loading, linking and deleting of separately linkedited user-written load modules. The subroutines thus referenced must be defined using the SUBMODS macro within the REENTSBS table.

The form of the MODCNTRL macro is as follows:

```
[symbol]  MODCNTRL  parameters
```

The parameters are as follows:

```
[link-parameters,]

ACTION={LOAD  }
       {LINK  }
       {DELETE}
       {SEARCH}

.MODNAME={module-name}
         {(r)       }

[.MF=(E,{link-param-list-addr})]
[       {(r)                 } ]

[.VL=VL]
```

link-parameters, MF, and VL are identical to the CALL macro parameter list specifications and cause a parameter list address to be passed to the load module in Register 1. These parameters are only valid if ACTION=LINK.

ACTION specifies the type of module-control request, as follows:

LOAD requests loading of the module. If the module is already loaded, its use count is incremented. All loads should be paired with deletes. The load module's entry point address is returned in Register 1.

LINK requests that the module be loaded, given control and deleted upon return (provided the use count is zero). The user may pass a parameter list to the load module by specifying "link-parameters" as described below.

DELETE requests that the module's use count be decremented, and, if it is zero, that the load module be deleted, based on the SUBMODS macro specifications.

SEARCH requests that the DYNLSUBS table, generated by the SUBMODS macro, be searched for the specified module. If found, the address at the DYNLSUBS entry is placed in register 0 and register 15 is set to zero. If not found, register 15 is set to minus 1.

MODNAME specifies the address of the eight-character EBCDIC left-justified load module name, padded with blanks.

MSGHDR -- Generate Message Header Fields

This macro (via COPY MSGHDRC) establishes symbolic names for the fields in the Intercomm 42-byte message header. This macro may be used in-line to generate the message header fields, or used within a Dsect when preceeded by a labeled DSECT statement.

The form of the MSGHDR macro instruction is as follows:

(blank)	MSGHDR	(blank)

PASS -- Transfer Ownership of an Area of Core from an Application to INTERCOMM

The PASS macro is used in a system with resource auditing and purging to protect an area of core acquired by an application thread from being freed by the purge routine when the thread completes.

[symbol]	PASS	ADDR={address / (r) / (1)} ,LEN={length / (r) / (0)} [,SPAEXT=(r)]

ADDR=
specifies the address of the area to be transferred; "address" may be any expression that can be substituted into the second operand field of an LA instruction. The address has to be doubleword aligned.

LEN=
specifies the length, in bytes, of the area to be transferred; "length" may be any expression that can be substituted into the second operand field of an LA instruction. The length will be rounded up to a multiple of 8.

SPAEXT=
specifies a register containing the address of the SPA Extension. If this operand is coded, the PASS macro will generate code to load the address of the routine that does the transfer from the SPA Extension. Otherwise, it will load a literal V-type constant.

(r) in the parameter descriptions means any number from 2 to 12, enclosed in parentheses; or any symbol equated to a number between 2 and 12, in parentheses.

PCENSCT

PCENSCT -- Print Accumulated Percentages Per Disk Queue

The PCENSCT macro instruction is used in conjunction with, and must follow, the SYCTTBL macro instructions. PCENSCT is used to print out percentages, via the SYCTTBL parameter PCEN=, that have accumulated for each queue, specified via the SYCTTBL parameter DFLN. Via this macro instruction, the user can determine what percentage of each disk queue has been allocated, and consequently how much of the queue is left for future allocations.

The form of the PCENSCT macro instruction is as follows:

[symbol]	PCENSCT	(blank)

PMISNAP -- Issue a SNAP

The PMISNAP macro should be used by subsystems in place of the IBM SNAP macro. It deducts the time taken by the SNAP operation from total elapsed time, thereby avoiding a subsystem timeout which could occur when taking a SNAP. The PMISNAP also provides other facilities described below.

The form of the PMISNAP macro is as follows:

[symbol]	PMISNAP	IBM SNAP Options: ID=n [,DCB=dcb-address] [,LIST=snaplist] [,MF={E,list / L}] [,PDATA=options] [,SDATA=options] [,STORAGE=storage-list] [,TCB=tcb-address] Intercomm SNAP Options: [,FAST={YES / NO}] [,INDUMP={YES / NO}] [, SPA / SPAEXT {=(r) / }]

The ID, DCB, LIST, MF, PDATA, SDATA, STORAGE and TCB parameters are specified exactly as they would be for a normal SNAP macro. Their usage and meaning are the same with the exceptions noted in the parameter descriptions below.

The FAST, INDUMP, SPA and SPAEXT operands do not apply to the IBM SNAP macro, or to the list form (MF=L) of the PMISNAP macro.

DCB

the user should code this parameter only if the snap is not to be routed to an Intercomm-administered snap data set, that is, the DD statements labeled SNAPDD or FASTSNAP. If a user-DCB is used, it is the user's responsibility to create and open the DCB.

FAST

NO specifies that a normal snap be issued. YES specifies that a special high-speed snap be taken. If FAST=YES is specified, the SDATA, PDATA, DCB, TCB, LIST and STORAGE operands are ignored and a full region dump is written to the data set defined by the FASTSNAP DD statement. If for some reason a fast snap cannot be performed, a slow snap will be issued. For this reason, it is recommended that FAST=YES be accompanied by "back-up" snap specifications, even through they are ignored when the fast snap is successful.

NOTE: For details on including the Fast Snap Facility within Intercomm (MVS only), see the Operating Reference Manual.

INDUMP

specifies whether this snap is eligible for the indicative snap option. INDUMP=NO specifies that the snap is ineligible. INDUMP=YES specifies that it is eligible. INDUMP=NO is the default. See the Operating Reference Manual for further details on the indicative dump option.

PDATA

in MVT, if the SPLS subparameter is coded, the Intercomm storage pools will be included in the snap. (This does not apply to MFT, since SPLS will automatically snap the entire partition.) If LIST or STORAGE was also specified, the Intercomm storage pools will appear in the snap before the user-specified areas. In this case, no more than 32 user-specified areas will be snapped.

SPA

specifies a register containing the address of the SPA. If this is coded, the PMISNAP macro will load the Intercomm snap routine address from the SPA Extension. Otherwise, a V-type address constant will be generated in-line to obtain the address. This parameter (or SPAEXT) must be coded when PMISNAP is issued by a module executing in the Link Pack area.

SPAEXT

specifies a register containing the address of the SPA Extension. If this is coded, the PMISNAP macro will load the Intercomm snap routine address from the SPA Extension. Otherwise, a V-type address constant will be generated in-line to obtain the address. This parameter (or the SPA parameter) must be coded when PMISNAP is issued by a module executing in the Link Pack area. The SPA and SPAEXT parameters are mutually exclusive. Only one or the other may be used.

PMISTOP -- Signal Logical End

The PMISTOP macro is employed in conjunction with a number of other Intercomm macros. Usually generating one fullword of hexadecimal FFs, various user-constructed tables require that this macro be used for signaling logical section or table ends.

NOTE: When the PMISTOP instruction is employed at the end of the BTERM instruction section of the Front End BTVRBTB table, it will, if there are graphic terminals defined to the system, also generate the graphic SAEC macro instructions.

The form of the PMISTOP macro is as follows:

(blank)	PMISTOP	(blank)

PMIWTO--Write to Operator

The PMIWTO macro generates a parameter list and calls the Intercomm module WTOMOD, which routes all WTOs according to user specifications. If a multiline WTO is issued, this macro should not be used; the IBM WTO macro should be used.

The form of the PMIWTO macro is as follows:

```
(symbol)  PMIWTO  {text                    }
                  {(text(,...,text))}

                  (,ID=aannn{I})
                  (         {A})

                  (,CSECT=xxxx)

                  (,DESC=number)

                  (,MF={L                    })
                  (    {(E,{list-address})}  )
                  (    {   {(1)          } } )

                  (,RENT={NO })
                  (      {YES})

                  (,ROUT={n})
                  (      {1})

                  (,ROUTCDE={route  })
                  (         {(2,11)})

                  (,ROUTX=(rout(,...,rout)))

                  (,{SPA   }=(r))
                  ( {SPAEXT}     )

                  (,TERM=terminal-id-address)
```

CSECT

specifies that a separate CSECT with name WTOxxxx be generated. All PMIWTO macros in the same module should be coded with the same identification. The CSECT generated may be assigned to an overlay region. Do not code this parameter if the SPA or SPAEXT parameters are used, or if the text of the PMIWTO is dynamically modified (contains variable text).

MF=

is the macro form following standard OS conventions. If MF=(E,list) is coded, list must be the label of a PMIWTOR macro coded with MF=L.

reply

defines the address for the reply message, per the standard IBM WTOR macro. This is a required positional parameter.

ROUT

must be coded as 1 (CPU console only).

ROUTCDE

specifies CPU console routing, per the standard IBM WTOR macro. The default is (2,11).

ROUTX

specifies an additional destination (message copy) via one or more subparameters, coded as follows:

OS--to CPU console (default)

SYSP--to SYSPRINT data set. The SYSPRINT DCB characteristics are assumed to include DCB=(BLKSIZE=141,LRECL=137,RECFM=VB,...)

CNTL--to the Intercomm control terminal

BROAD--to the broadcast group TOALL

EXIT--to a user exit with the CSECT name USRWTO. (Refer to Messages and Codes.)

The ROUTX specification does not negate the ROUT parameter, but may be used in addition to or instead of the ROUT specification.

SPA

specifies the address of the System Parameter Area (SPA) in register notation. This causes the WTOMOD module address to be obtained from the SPA Extension, rather than from an assembled-in VCON, as would be necessary if the module containing the PMIWTOR is linkedited separately from WTOMOD. This parameter is needed if WTOMOD is not in the resident portion of Intercomm and the module issuing the macro is dynamically loadable.

SPAEXT

specifies the address of the SPA Extension (SPAEXT) in register notation. This causes the WTOMOD module address to be obtained from the SPA Extension rather than from an assembled-in VCON, as would be necessary if the module containing the PMIWTOR is linkedited separately from WTOMOD. This parameter is needed if WTOMOD is not in the resident portion of Intercomm and the module issuing the macro is dynamically loadable. The SPA and SPAEXT parameters are mutually exclusive. Only one or the other may be coded.

text

denotes the fixed and/or variable text to be issued via the standard IBM WTOR macro (with the usual length restrictions imposed by the operating system in use), and is one of the forms described for the PMIWTO macro.

NOTE: For additional information on the use and coding of the keyword parameters, see the PMIWTO macro. When using list and execute forms of PMIWTOR, reply, len and ecb must be coded on the execute form.

DESC

defines the message descriptor code, per the standard IBM WTO macro.

ID

provides the last six characters of a nine-character Intercomm message identifier. aa must be alphabetic; nnn must be numeric. I denotes an informational message; A denotes that some corrective action should be performed. The first three characters form the regionwide id prefix specified via the SPALIST parameter WTOPFX (default is INT). If this parameter is not coded, a nine-character identifier is assumed to be present at the beginning of the text.

MF

is the macro form following standard OS conventions. If MF=(E,list-address) is coded, list-address must be the label of a PMIWTO macro coded with MF=L, and specifying the following parameters (as desired): text, ID, DESC, ROUT, ROUTX, ROUTCDE, TERM.

RENT

indicates whether the issuer of the PMIWTO macro is quasi-reentrant. If RENT=NO is coded, then it is assumed that register 13 does not point to an available register save area and ROUT=1 is forced; all ROUT and ROUTX specifications are ignored. If RENT=YES is specified, all ROUT or ROUTX specifications are honored; register 13 is assumed to point to a standard save area. The default is YES.

ROUT

specifies the receiving devices, as follows:

1. CPU console only (default).
2. CPU console and Control Terminal.
3. Control Terminal only.
4. CPU console and Broadcast Group TOALL.
5. User Exit (USRWTO - see ROUTX).
6. CPU console and SYSPRINT.
7. CPU console, SYSPRINT, and Control Terminal.
8. CPU console: only if WTO is coded as an Intercomm EXEC parameter.

ROUTCDE

specifies CPU console routing, per the standard IBM WTO macro. The default is (2,11). The SPALIST parameters FMCSWTO and SMCSWTO can be used to add or delete specific ROUTCDE route codes on a regionwide basis.

ROUTX

specifies the destination via one or more subparameters, coded as follows:

OS--to CPU console

OSIF--to CPU console, if WTO is specified as an Intercomm EXEC statement parameter

SYSP--to SYSPRINT data set. The SYSPRINT DCB characteristics are assumed to include DCB=(BLKSIZE=141,LRECL=137,RECFM=VB,...)

CNTL--to the Intercomm control terminal

BROAD--to the broadcast group TOALL

EXIT--to a user exit with the CSECT name USRWTO. (Refer to Messages and Codes.)

TERM--to the terminal specified in the TERM parameter

The ROUTX specification does not negate the ROUT parameter, but may be used in addition to or instead of the ROUT specification.

The SPALIST parameters FPMIWTO and SPMIWTO can be used to add or delete specific ROUTX route codes on a systemwide basis.

SPA

specifies the address of the System Parameter Area (SPA) in register notation. This causes the WTOMOD module address to be obtained from the SPA Extension, rather than from an assembled-in VCON, as would be necessary if the module containing the PMIWTO is linkedited separately from WTOMOD. This parameter is needed if WTOMOD is not in the resident portion of Intercomm and the module issuing the macro is dynamically loadable.

SPAEXT

specifies the address of the SPA Extension (SPAEXT) in register notation. This causes the WTOMOD module address to be obtained from the SPA Extension rather than from an assembled-in VCON, as would be necessary if the module containing the PMIWTO is linkedited separately from WTOMOD. This parameter is needed if WTOMOD is not in the resident portion of Intercomm and the module issuing the macro is dynamically loadable. The SPA and SPAEXT parameters are mutually exclusive (only one or the other may be coded). When using list and execute forms of PMIWTO, SPA/SPAEXT must be coded on the execute form.

TERM

specifies the address of a field which contains the five-character terminal-ID of a terminal to receive this message. This address is placed in register 2 at execution time. If list and execute forms of the macro are used, this parameter must be coded on both.

text

denotes the fixed and/or variable text to be issued via the standard IBM WTO macro (with the usual length restrictions imposed by the operating system in use), and is of one of two forms:

```
{'actual-text'                          }
{(name,length[,fill-character])}
```

where:

- actual-text is a fixed literal string enclosed in quotes.
- name specifies a one- to eight- character symbolic field name.
- length specifies the length attribute of the alphanumeric field to be generated.
- fill-character specifies the initialization character to be generated. Default is a blank.

For example:

PMIWTO ('AAAA',(X11,5),'BBB',(X22,12,*)) generates the following storage text definitions:

```
         DC     C'AAAA'                 fixed literal
X11      DC     CL5' '                  variable text
         DC     C'BBB'                  fixed literal
X22      DC     CL12'************'       initialized variable
```

The programmer is thus provided a label reference by which to vary all or portions of write-to-operator message texts. The generated message text length may not be greater than 122 characters if the ID parameter is coded, nor greater than 132 if the message id is in the text string.

All messages routed to the CPU console are prefixed by an eight-byte region (job-name) id and the nine-byte message id (if ID parameter coded). Additionally, messages routed to SYSPRINT are prefixed by a twelve-character time stamp.

PMIWTOR--Write to Operator with Reply

The PMIWTOR macro generates a parameter list and calls the module WTOMOD, which routes all WTOs and WTORs according to user specifications. The program issuing the macro may regain control after the reply takes place by issuing a DISPATCH macro to wait on the ECB specified in the macro parameters. If a multiline WTOR is used, this macro should not be used; the IBM WTOR macro should be used instead.

The form of the PMIWTOR macro is as follows:

```
(symbol)  PMIWTOR  text,reply,len,ecb
                   (,ID=aannnR)
                   (,DESC=number)
                   (,MF={L                    })
                   (     {(E,{list-address})} )
                   (     {   {(1)         } } )
                   (,ROUT=1)
                   (,ROUTCDE={route })
                   (         {(2,11)} )
                   (,ROUTX=(rout(,...,rout)))
                   (,{SPA   }=(r))
                   ( {SPAEXT}     )
```

DESC

defines the message descriptor code, per the standard IBM WTOR macro.

ecb

defines the address of the ECB to indicate completion of the reply, per the standard IBM WTOR macro. This is a required positional parameter.

ID

provides the last six characters of a nine-character Intercomm message identifier. aa must be alphabetic; nnn must be numeric; R denotes an operator reply is being awaited.

len

defines the length of the reply message, per the standard IBM WTOR macro. This is a required positional parameter.

POLLIST -- Define Polling List (BTAM frontend)

The POLLIST macro instruction is used to specify to INTERCOMM's BTAM frontend the polling list that is to be associated with a specific polled line. Refer to the BLINE macro "POLLIST=" parameter. In essence, the POLLIST macro instruction generates a BTAM DFTRMLST* macro instruction but it further functions to prcvide the frontend with required associated internal table displacements.

The form of the POLLIST macro instruction is as follows:

symbol	POLLIST	list-type,line, EXTNAM= {(term-id[,term-id],...) {term-id CTCHAR= {(poll-chars[,poll-chars],...) {poll-char CONTIN={YES {NO

symbol
supplies a label by which the pollist list can be identified. This identification is required in order for the list to be associated with a specific line. Refer to the BLINE macro "POLLIST=" parameter.

list-type
specifies the format of the polling list. Code one of the following symbols:

OPENLST - specifies the generation of an open polling list.
WRAPLST - specifies the generation of a wraparound polling list.
SSALST - specifies the generation of an open polling list for Auto poll operations on start-stop lines.
SSAWLST - specifies the generation of a wraparound list for Auto poll operations on start-stop lines.
AUTOLST - specifies the generation of an open polling list for Auto poll operations on bisync lines.
AUTOWLST- specifies the generation of a wraparound polling list for Auto poll operations on bisync lines.

NOTE: See also AUTO= parameter on BLINE macro.

The code assigned to this parameter will be substituted onto the "list type" positional parameter of a DFTRMLST macro instruction.

---

*Refer to IBM SRL 'BTAM' document GC30-2004.
,... designates that multiple iterations of the sub-list elements "term-id" and "poll-chars" may occur.

line
designates the specific line to which this polling list is to be applied. This line must be designated by a code constructed in the following manner: by taking the label identifying the LINEGRP macro instruction within whose group of lines the subject line belongs by concatenating to the end of this label the letter 'L', and then by concatenating to the end of this result the relative line number associated with the line. (This line number must not possess leading zeros.) The relative line number is determined as follows: the LINEGRP macro instruction subordinates one or more BLINE macro instructions and a number (relative to 1) is implicitly assigned to each BLINE macro instruction by coded sequence alone, i.e., the second coded BLINE macro instruction representing a member line within the line group is implicitly assigned a 2. This number is the relative line number. For example: A code of "S741L8" would be created from a LINEGRP label of "S741" and a relative line number of 08.

EXTNAM=
specifies the INTERCOMM terminal-id's of all terminals to be polled on the line. These terminal-id's must be identical to those already declared within the group of BTERM instructions that represent the set of terminals associated with the line. Any one terminal name may be repeated up to eight times on this parameter, though no more than 31 terminal names are permitted to be specified. If more than one terminal name is to be supplied the names must be coded as subparameters within a parameter sublist.

CTCHAR=
specifies the polling characters to be employed when the polling operation is performed upon the associated line. There must be a one-to-one mapping between the terminal name (s) supplied by the "EXTNAM=" parameter and the polling code(s) to be used with those terminals as they are supplied by this parameter. Note that this necessarily means that if a terminal name is repeated within an "EXTNAM=" sublist, the appropriate polling characters must likewise be repeated. Each polling character must be rendered into the appropriate transmission code for the specific device type and coded as a pair of hexadecimal digits. The code assigned to this parameter will be substituted onto the "device dependent operand" positional parameter of a DFTRMLST macro instruction.

CONTIN=
specifies that the remainder of the polling list can be found in succeeding POLLIST macros. This would be necessary whenever the CTCHAR or EXTNAM parameters exceed

255 characters (including commas and parentheses), the assembler limit for a macro instruction operand. To use this parameter, code the first POLLIST macro normally with CONTIN=YES specified. Code all following POLLIST macros, except the last, with CONTIN=YES. Note that the POLLIST macro label, list-type, and line parameters may be omitted on all but the first POLLIST macro. The default value for this parameter is NO.

REGA -- Equate GP Registers (hexadecimal, decimal)

The REGA macro instruction generates a set of twenty-two EQU assembler instructions. These are of the following form:

```
Rm  EQU  n
```

where "m" is either a hexadecimal value 0-F or a decimal value 10-15, and "n" is its decimal equivalent (cf. REGS instruction. The labels so generated are intended to function as general register symbols.

The form of the REGA macro instruction is as follows:

(blank)	REGA	(blank)

REGS -- Equate GP Registers (decimal)

The REGS macro instruction generates a set of sixteen EQU assembler instructions. These are of the following form:

```
Rn  EQU  n
```

where "n" is a decimal value 0-15 (cf. REGA macro). The labels so generated are intended to function as general register symbols.

The form of the REGS macro instruction is as follows:

(blank)	REGS	(blank)

RESOURCE -- Specify a Shared Resource

The RESOURCE macro is used in conjunction with the SYCTTBL macro to limit the number of threads that may concurrently access a specific system resource. RESOURCE macros must be coded prior to any SYCTTBL macros.

The form of the RESOURCE macro is as follows:

```
symbol    RESOURCE   parameters
```

The parameters are as follows:

```
ID=resource-ID


[,MAXUSE={n}]
[        {1}]
```

ID specifies the resource, via an identifier of eight characters or less. This identifier is enqueued upon by the Subsystem Controller before calling the related subsystem (see SYCTTBL macro RESOURC parameter) in order to limit the maximum number of concurrent threads.

MAXUSE specifies the maximum number of threads allowed to use this resource at any one time. The default is 1.

ROUND -- Resolve Upwards (power of 2)

The ROUND macro instruction performs an upwards (power of 2) resolution upon a value contained within a designated register. The precision of the resolution is variable and the result is returned in the same register the original value was given in. All other registers remain unchanged.

The form of the ROUND macro instruction is as follows:

[symbol]	ROUND	register [,P= {n / 3}]

register
designates the general register (1-15) that contains the value to be operated upon. Note that bits 8-31 only are considered valid and the particular ceiling value above which ROUND'ing will be erroneous is expressible as follows:

$$(2^{23}-1)-(2^{n}-1)$$

where "n" is the value provided by "P=". Upon return, bits 0-7 are zero.

P=
specifies the 'n'th power of 2 up to whose initial comparatively higher multiple the subject value is to be raised. The default code is 3, providing resolution upward to a multiple of 8. Code as a decimal value, 2 to 8.

RTNLINK -- Restore Registers, Free Save Area, Return

The RTNLINK macro, coded in coordination with the LINKAGE macro, restores the registers saved at LINKAGE time, optionally frees any main storage acquired at LINKAGE time, and returns to the CALLing program with a optional return code.

The form of the RTNLINK macro is as follows:

```
[symbol]  RTNLINK  [ADDR={(r)  }]
                   [     {(13)}]
                   [,FREE={YES}]
                   [      {NO }]
                   [,GEN={NO  }]
                   [     {YES }]
                   [,LEN={length}]
                   [     {(r)   }]
                   [     {(0)   }]
                   [,PRELOAD={YES}]
                   [         {NO }]
                   [,RC={return-code}]
                   [    {(r)       }]
                   [    {(15)      }]
                   [,SPA=(r)]
```

ADDR=

provides the address of the main storage (save/work) area, the first 72 bytes of which have been used as the program's save area. Note that though 'r' may designate any general register, if Register 13 has not been disturbed since LINKAGE issuance, 13 should be coded.

FREE=

designates explicitly whether or not the main storage area is to be freed. Code YES if it is to be freed; code NO if it is not. If neither a YES nor a NO is coded, the storage will be implicitly freed if the preceding LINKAGE instruction specified DYN=YES, or implicitly not freed if it specified DYN=NO.

GEN=

specifies whether or not the PMIRTLR CSECT is to be generated (valid only for the first RTNLINK macro in a program). Code YES to generate PMIRTLR CSECT; code NO for suppression. The default is YES.

LEN

provides the length, in bytes, of main storage to be freed. 'length' may be any expression that can be substituted into the second operand field of an LA instruction; for example, 2108, EQUATE, 0(2,4). Note that normally the value assigned to this parameter would be identical to that assigned to the LEN parameter of the LINKAGE macro. This parameter is not to be coded if DYN=NO is coded on the preceding LINKAGE instruction. It is required under all other circumstances.

PRELOAD

specifies whether or not VS page preloading is to be utilized. If YES is coded, it requests that if the Intercomm system is being run under VS then the pages containing the higher-level save area and the return point of the calling module are to be preloaded into real storage before they are referenced to prevent possible page-faults. The default is NO.

RC

provides the return code to be passed to the calling program. 'return-code' may be any expression that can be substituted into the second operand field of an LA instruction. The permissible return code range is from 0 to 8,388,607; however, if a return is being made to Intercomm's Subsystem Controller, the usable return codes may be considered to be limited to the 0 to 255 range. This limitation is imposed since the Subsystem Controller uses a one-byte message header field (MSGHRETN) in which to record the code before logging the header. Intercomm return code conventions to be used when control is being returned to the Subsystem Controller are described in the Assembler Language Programmers Guide.

SPA

provides the register containing the address of the System Parameter List. This parameter need be coded only if a previous LINKAGE macro does not specify the SPA parameter or coding logic is such that the previously indicated SPA base register is not valid at the time this RTNLINK is executed.

SECVERBS -- List Transaction-IDs Requiring Security

The SECVERBS macro instruction is used in conjunction with the GENSEC and STATION macro instructions. These three instructions are employed together to create the PMISTATB and PMISECTB table entries that furnish the INTERCOMM Output, RJE and Security subsystems with the terminal information they require. The SECVERBS macro instruction is required and meaningful only when INTERCOMM's transaction-ID terminal security is utilized and the instruction is employed specifically to make known to the security subsystem all transaction-IDs that are expected to undergo a terminal security check. If the verb sublist contains more than 255 characters, more than one SECVERBS macro should be coded, one after the other and these must precede all STATION macro instructions assembled with it.

The form of the SECBERBS macro instruction is as follows:

(blank)	SECVERBS	VERBS=(trans-id[,trans-id],...) [,TABLE={YES / NO}]

VERBS=

specifies the one or more transaction-IDs that are to be provided with terminal security. The maximum number of characters permissible within the parameter sublist is 255. If TABLE=NO is specified, the order of the transaction-IDs is critical; if TABLE=YES is specified, it is not. Refer to the note under the TABLE= parameter. Refer also to the STATION macro VERBS= parameter.

TABLE=

specifies whether or not an in-line table consisting solely of the transaction-IDs supplied by the VERBS= parameter is to be generated. If more than one SECVERBS macro is coded, this parameter must be coded identically on each. Code YES if an in-line table is to be generated. Code NO if an in-line table is not to be generated. If NO is coded, the Intercomm BTAM Front End BTVRBTB table must exist, and must conform to the restriction specifically noted below. In this instance, the BTVRBTB table is used as a substitute for the in-line table that would otherwise be generated.

NOTE: If NO is coded, the order of transaction-IDs given via the VERBS= parameter must be in the identical

order of the BTVERB macro instructions defining those transaction-IDs within the BTVRBTB table; furthermore, all BTVERB macro instructions defining these transaction-IDs must precede any BTVERB macro instruction that defines a transaction-ID not requiring a security check.

SPALIST -- Supply System Specifications

The SPALIST macro is used to provide certain required system information. Much of this information, such as subroutine addresses, table addresses, reserved codes, equates, is unchangeable and automatically supplied by the macro; however, some of the information required (environmental, procedural, timing) is changeable and supplied via the parameters specified below. This information is stored in the Csects SPA and SPAEXT. Only one SPA and one SPAEXT are permitted to be defined in each Intercomm region. All SPALIST information is implicitly made accessible to all subsystems, which receive the address of the SPA as the second fullword of the passed parameter list. (Refer to the LINKAGE macro, SPA parameter and the DSECTS parameter, suppression symbol SPA.)

In addition, the SPA Csect is automatically followed by a user area called the userspa, if a partitioned data set COPY member named USERSPA is provided (at SPA assembly time). This permits the user to add to the SPA Csect whatever systemwide information or fields may be uniquely required to be accessible to all user programs.

The System Parameter Area (SPA Csect) and the Intercomm SPA Extension (SPAEXT Csect) are generated by coding the module INTSPA containing the SPALIST macro with the EXTONLY=BOTH parameter (and other applicable parameters), and followed by an END statement.

The form of the SPALIST macro is as follows:

(blank)	SPALIST	macro qualifier parameters: A={A} {B} [,EXTONLY={BOTH}] [{YES }] [{NO }]

(continued)

environment parameters:

```
[,AUTOMXF={max-num-autogen-fields}]
[         {100                   }]

[,AUTOPCH={ddname-of-autogen-punch-file}]
[         {AUTOGPCH                    }]

[,AUTOSFD={autogen-store/fetch-ddname}]
[         {INTSTOR0                  }]

[,MRAUTO={WAIT}]
[        {ENDN}]
[        {DUMP}]
[        {NO  }]

[,MRCNTL={YES}]
[        {NO }]

[,MRCSALN={multi-region-msg-area-len}]
[         {1024                     }]

(,MRID=satellite-region-id)

[,MR1LOG={YES}]
[        {NO }]

[,TOTATT={NO }]
[        {YES}]

[,TPMOD={T}]
[       {A}]
[       {F}]

[,TSTEND= DUMP  }]
[        {NODUMP}]
[        {NRCD  }]
```

execution parameters:

```
[,ASYNLDR={NO }]
[         {YES}]

[,CCNID={intercomm-control-tid}]
[       {CNT01                }]

[,CLDNLIM={n  }]
[         {300}]

[,CLDTO={DUMP  }]
[       {NODUMP}]
```

(continued)

```
[,CNTDT={resident-workscan-interval}]
[       {1                         }]

[,DTIMS={terminal-busy-delay}]
[       {4                  }]

[,DWSCHK={YES}]
[        {NO }]

[,ECB={NO }]
[     {YES}]

[,EDITRTN={num-of-edit-subroutines-in-use}]
[         {20                             }]

[,GPSSEC=FEOVLOG]

[,INDUMP={YES[,{n}]}]
[        {     {4} }]
[        {NO       }]

[,LOGINDO={YES}]
[         {NO }]

[,MBPR={min-size-blk-subpool-0-core}]
[      {2048                       }]

[,MDELY={max-dely-time-value}]
[       {240                }]

[,MMNCL={max-mncl-value}]
[       {10            }]

[,MSPR={min-available-subpool-0-core}]
[      {2048                        }]

[,NQTIM={enqueue-default-time-intvl}]
[       {60                        }]

[,NTIMS={max-num-attempts (core,etc.)}]
[       {7                          }]

[,NWAIT={max-num-icom-waiting-msgs}]
[       {0                        }]

[,SEP={separator-char}]
[     {6B            }]

[,SNAPPGS={num-of-snap-pages-before-spinoff}]
[         {0                               }]
```

(continued)

```
[,SSNUM={max-num-start/stop-functions}]
[       {240                         }]

[,STOCORE={store/fetch-core-allotment}]
[         {5                         }]

[,STSTIME={system-tuning-statistics-intvl}]
[         {120                             }]

[,STUSPIE={interruptions}]
[         {((1,13),15)  }]

[,SWIN={ovly-reg-workscan-interval}]
[      {1                          }]

[,TASKNUM={(num-of-general-subtasks,num-of-special-subtasks)}]
[         {(1,0)                                            }]

[,TIMS={core-acquisition-delay}]
[      {2                     }]

[,TITECOR={YES}]
[         {NO }]

[,WTO={NO }]
[     {YES}]
```

checkpoint parameters:

```
[,CKPTLIM={checkpoint-time-limit}]
[         {60                   }]

(,CKUSR=user-ckpt-area,CKUSL=length-user-ckpt-area)

[,GENSW={disk-recording-area-bits}]
[       {7C                      }]

[,TCHP={checkpoint-interval}]
[      {120                }]
```

security parameters:

```
[,SGNTIME={sign-off-default-time-intvl}]
[         {5                           }]

[,SONOFF={YES}]
[        {NO }]

[,TRANSEC={YES}]
[         {NO }]
```

(continued)

```
[,USERSEC={NO }]
[         {YES}]
```

resource management parameters:

```
[,COREACC={NO }]
[         {YES}]

[,CUSHION={storage-cushion-size}]
[         {2048                }]

[,CUSHTM={cushion-reacq-intvl}]
[        {1                  }]

[,RCBSADD={rcb-num-increment}]
[         {5                }]

[,RCBSINT={initial-num-of-rcbs}]
[         {75               }]

[,RMSTIM={resource-statistics-time-interval}]
[        {5                                }]

[,TRACETM={core-use-statistics-printout-intvl}]
[         {120                               }]
```

dynamic subsystem loading specifications:

```
[,MAXLOAD={max-subpool-space}]
[         {50000            }]
```

message logging specifications:

```
[,LGBLK={interlog-buffer-length}]
[       {2000                  }]

[,LGNUM={num-of-interlog-buffers}]
[       {2                     }]
```

PMIWTO(R) specifications:

```
[,FMCSWTO=additional-route-codes]

[,FPMIWTO=additional-route-codes]

[,SMCSWTO=excludable-route-codes]

[,SPMIWTO=excludable-route-codes]

[,WTOPFX={intercomm-wto-prefix}]
[        {INT                 }]
```

(continued)

VSAM local-shared-resources specifications:

```
[,BLDVRP={YES}]
[        {NO }]

[,BUFFERS=(size(n)[,...,size(n)])]

[,FIX={(BFR,IOB)}]
[     {(BFR)    }]
[     {(IOB)    }]

[,KEYLEN={n}]
[        {1}]

[,STRNO={n}]
[       {1}]
```

In the following parameter descriptions, one or more capitalized symbols in parentheses refer to labels within the SPALIST DSECT directly associated with the parameter. For example, an expression such as

(SPABITS(SPAASYN,YES=1,NO=0))

means that if NO is coded on the associated parameter, the bit identified by the equate label SPAASYN, and residing within the field identified by the label SPABITS, is set to 0. One or more symbols parenthetically qualifying another imply relevant equated bits within a field.

A

indicates whether the SPALIST macro is being used to generate a CSECT or a DSECT. Code A=A if a CSECT is desired. The default is B, specifying that a DSECT is to be generated: requires a labeled DSECT statement preceding the macro, which is the object of a USING statement.

ASYNLDR

specifies whether or not Intercomm is to perform asynchronous overlay loading. The default is YES. (SPABITS(SPAASYN,YES=1,NO=0)).

AUTOMXF

is a required parameter for Autogen users. It indicates the maximum number of fields that are to be created in any execution of Autogen. Code as a decimal number in the range 1 to 256, inclusive. The default is 100. (SEXMXFLD)

AUTOPCH
is a required parameter for Autogen users. It specifies the ddname of the file that receives the macros that define the generated screen. The default is AUTOGPCH. (SEXAUTPC)

AUTOSFD
is a required parameter for Autogen users. It specifies the ddname of the Store/Fetch file to be used by Autogen. It must be in the form INTSTORx, where x is in the range of 0 to 9 inclusive. The default is INTSTOR0. (SEXAUTSF)

BLDVRP
specifies whether or not a VSAM local shared resources pool is to be built. If NO is coded, no resource pool is built and, if LSR is coded on a FAR statement, it is ignored. If YES is coded, a parameter list for the BLDVRP macro is created in the Csect VRPLIST and the resource pool will be built at execution time. If YES is coded, the BUFFERS, KEYLEN, STRNO and FIX parameters may need to be coded. If NO is coded, these parameters are ignored. For further information, see the Operating Reference Manual. Also, consult IBM's VSAM Options for Advanced Applications. The default is NO.

BUFFERS
specifies buffer pool configurations for the VSAM local shared resources pools. Code size and number exactly as it would be for a BLDVRP macro. Note that number must be at least 3, size must be the CISIZE from a LISTCAT of each file (component) to use the buffer pool. (See VSAM Options for Advanced Applications.) There is no default for this parameter. If BLDVRP=YES, this parameter is required.

CCNID
specifies the terminal-ID assigned to the control terminal. The value coded must be identical to the value coded for the CONTROL parameter of the associated BTERM macro and the value defined for SETENV symbolic variable &CNTL. The default is CNT01. (SPACCNID).

CKPTLIM
specifies, in seconds, the maximum length of time to allow for a checkpoint to complete when used in conjunction with File and/or Data Base Restart/Recovery. At the expiration of the time interval, message number DB674I is issued, all quiesced subsystems are reactivated, and the checkpoint is skipped. Care should be taken that this value is not in conflict with the TCTV parameter in the checkpoint subsystem SYCTTBL macro. Code as a decimal value in the range of 10 to 55924, inclusive. The default value is 60 seconds. (See also TCHP parameter.) (SEXCPLIM)

CKUSL

specifies, in bytes, the length of the user area that is to be checkpointed. This parameter is meaningful only if CKUSR has been specified. Code as a decimal value (range 1 to 32767), as a hexadecimal self-defining term, or as a term specifying the length attribute of an area-associated symbol. (SPACKUSL)

CKUSR

specifies, via a V-type or an A-type address constant, the address of the user area that is to be checkpointed. Code a V-type address constant if the area is not assembled within the same module as the SPALIST instruction. Code an A-type address constant if the area is assembled within the same module (USERSPA). If this parameter is specified, CKUSL must also be coded. (SPACKUSR)

CLDNLIM

specifies, in seconds, the maximum length of time to allow for a closedown to complete. At the expiration of the time interval, no further messages are processed, and no further output is sent. Closedown immediately terminates all Intercomm functions and returns to the operating system. The default is 300 seconds. (SEXCDLIM)

CLDTO

specifies whether or not an abend is to be performed when a closedown time-out occurs (see also CLDNLIM). If NODUMP is coded, an immediate closedown is done; if DUMP is coded, an Abend 125 with dump option is issued. The default is NODUMP. (SPABITS3(SPACLDMP,NODUMP=0,DUMP=1))

CNTDT

specifies, in seconds, the duration of time allowed by the Subsystem Controller to elapse between the initiation of one scan to detect any work queued for resident, or overlay region B, C, or D Subsystem Control Table entries, and the initiation of the next such scan. This value is meaningful only for those entries whose associated SYCTTBL macro has specified ECB=NO. Code as a decimal value, 1 to 27962. The default is 1. (See also SWIN parameter.) (SPACNTDT)

COREACC

specifies whether or not detail core-use statistics are to be accumulated. If YES is coded, the detail statistics are headed "Distribution of Core Block Sizes" and "Pool Use Detail Statistics" in the core use statistics printout. The parameter is meaningful only if the resource management module (MANAGER) has been assembled with the SETGLOBE global &RMACCT set to 1. The default is YES. (See also RMSTIM and TRACETM parameters.) (SEXRMFLG(SEXSTAC,YES=1,NO=0))

CUSHION

specifies the size, in bytes, of a block of storage to be acquired via an unconditional GETMAIN at startup, and to be released in the event of a storage request that cannot be satisfied from the user-defined pools or from OS-administered dynamic storage. An operator message (RM019I) is issued when the cushion is released. Code as a decimal multiple of 2048. The default is 2048. (SEXCUSH)

NOTE: The size value should be coded as a multiple of the OS subpool block or VS page frame size (2-4K) as applicable under the operating system in use. Once the Intercomm core pools are tuned so that all system work area, message area, dynamic save/work area, etc., requests are normally filled from the pools, then a size in the range 4-12K should be sufficient. Under MVS with a large virtual region size, the CUSHION value may be 0.

CUSHTM

specifies the interval, in seconds, between attempts to reacquire the storage cushion, once it is released. An operator message (RM020I) is issued when the cushion is reacquired. Code as a decimal value in the range 0 to 109 inclusive. The default is 1. (SEXRETRY)

DTIMS

specifies, in seconds, the duration to wait before another attempt is made to check the availability of a terminal after a device-busy condition has been detected as existing for that terminal. This parameter is meaningful only if segmented output message processing is requested. The device busy condition exists from the moment the Output Utility receives the initial segment of a segmented message (MSGHVMI=X'51') destined for that terminal, through to the moment that the Output Utility completes its processing of the last segment of that segmented message. Code as a decimal value, 1 to 27962. The default is 4. (See also TIMS and NTIMS parameters.) (SPADTIMS)

DWSCHK

specifies whether the COBOL Dynamic Working Storage protection option is in use. This permits checking if a DWS specification is too small. If YES is coded, an additional pad area is appended to the dynamic working storage area which is inspected on each call to COBREENT. If it has been modified, the thread is terminated, a diagnostic message is issued and a snap 126 is taken. The default is NO. DWS checking may be dynamically controlled via the STRT/STOP system commands. (SEXBITY(SEXDWS,YES=1,NO=0))

ECB

indicates whether or not the Subsystem Controller is to await notification of the arrival of work upon a queue within an overlay region A Subsystem Control Table entry via the posting of an ECB. For an equivalent feature for a resident, or overlay region B, C, or D entry, refer to the SYCTTBL macro, ECB parameter. Code YES if the Subsystem Controller is to await notification via a posted ECB. Code NO if the Subsystem Controller is alternatively to regularly initiate a scan for arrived work. The subject ECB resides in the System Parameter Area and is identified by the label SPACTIVE. The default is YES. (See also SWIN parameter.) (SPABITS1(SPAECBWT,YES=1,NO=0))

EDITRTN

specifies the number of the highest-numbered edit subroutine in use. Edit subroutines must have names of the form EDITnnn where nnn is a three-digit number. The default code is 20. (EDITRTNS CSECT in SPA Extension.) Refer to the Operating Reference Manual.)

EXTONLY

specifies, if A=A is coded, if both the SPA and SPAEXT Csects are to be generated, or if only the SPA Csect or the SPAEXT Csect should be generated. Code BOTH if both Csects are to be generated in one module (INTSPA). Code YES if only the SPAEXT Csect is desired. The default is NO (only SPA Csect generated).

FIX

specifies that the VSAM LSR buffer pool (BFR), I/O related control blocks (IOB) or both (BFR,IOB) are to be page-fixed in real storage. The default is null (that is, neither item page-fixed.) If page fixing of resource pool elements is requested, the Intercomm Interregion SVC must be installed in the system. (See also BLDVRP parameter.)

FMCSWTO

lists ROUTCDE route codes (see PMIWTO and PMIWTOR macros) to be added to all PMIWTO and PMIWTOR executions. The route code 8 is automatically included unless 8 is listed in the SMCSWTO parameter. (SEXFMCS)

FPMIWTO

lists ROUTX route codes (as described in PMIWTO and PMIWTOR macros) to be added to all PMIWTO and PMIWTOR executions. (See also SPMIWTO parameter.) (SEXFPMI)

GENSW

specifies, by bit association, which of five checkpoint recording segments (of the CHEKPTFL data set) are to be considered usable at startup time. A bit set to 1 designates that the associated segment is not usable. A bit set to 0 designates that the associated segment is usable. The five checkpoint areas are represented by bits 1 through 5 of the byte code assigned to this parameter. Code as two hexadecimal digits. If more than three of the five bits are set to 1, no checkpointing will be done. The default is 7C (no checkpointing). See the Operating Reference Manual. (SPAGENSW)

GPSSEC

restricts selected GPSS functions to the control terminal only. FEOVLOG value applies to the FILE system control command. If coded, an FEOV request for INTERLOG is restricted to the control terminal. (SPAGPSEC(SPAEVLOG,YES=1,NO=0))

INDUMP

specifies whether the indicative snap option is desired. YES specifies indicative snaps, which causes eligible snaps (as determined by the INDUMP parameter of the PMISNAP macro) to produce only those storage areas most needed for debugging purposes. This minimizes printing and disk space requirements for the affected snaps. Intercomm 114, 118 and 126 snaps are automatically eligible for this option. The second subparameter specifies the portion of nonreentrant subsystems to be snapped, in kilobytes. For example, code INDUMP=(YES,10) if only the first 10K of all nonreentrant subsystems is desired. If the second subparameter is omitted, 4 (K) is the default. Code as a decimal value from 0 to 255 (SEXDMLN). NO is the default. See the Operating Reference Manual for further details on the indicative dump option. INDUMP processing may be dynamically controlled via the STRT/STOP system commands. (SEXBITA(SEXDMP,YES=1,NO=0)

KEYLEN

specifies the maximum key length of any VSAM data set that will use the local shared resources pool. Code as a decimal value from 1 to 255. The default is 1. (See also BLDVRP parameter.)

LGBLK

specifies, in bytes, the expected average length of INTERLOG buffers. (INTERLOG records are variable-length; maximum buffer length is 32K). Code as a decimal value from 48 to 32764. The default is 2000. (SEXLGBLK)

LGNUM

specifies the number of INTERLOG buffers. Code as a decimal value. The specified value must match the value of the NCP subparameter of the DCB parameter of the INTERLOG DD statement. These buffers are obtained at startup time and retained for the duration of the Intercomm run. The default specification is 2, that is, two buffers each of the size specified by the LGBLK parameter. (SEXLGNUM)

LOGINDO

specifies whether or not terminal output generated by the LOGINPUT Facility (see the Operating Reference Manual) is to be discarded. If YES is coded, then the output will be discarded by substitution of the &GENTERM specification (defined in SETGLOBE) in the LOGINPUT message header field, MSGHTID. (The terminal-ID defined by &GENTERM must be defined via a STATION macro.) NO is the default. (SPABITS2(SPALINDO,YES=1,NO=0))

MAXLOAD

specifies, in bytes, the maximum allowable total subpool space to be occupied at any time by dynamically loaded subsystems. Code as a decimal value; the default is 50000. The coded value may be dynamically overridden via the LOAD command, CORE parameter (see System Control Commands). (SEXSPMAX)

MBPR

specifies, in bytes, the minimum size of a single contiguous block of subpool 0 core that must be available before a subsystem can be forwarded a message. Code as a decimal value, 0 to 32760. The value assigned to this parameter must be equal to or less than that assigned to the MSPR parameter. See NOTE under CUSHION parameter. The default is 2048. (SPAMBPR)

MDELY

specifies, in minutes, the maximum time delay value that may be entered on an Intercomm DELY system control command. Code as a decimal value, 2 to 466. The default is 240 (4 hours). (SPAMDELY)

MMNCL

specifies the maximum SCTMNCL value (refer to SYCTTBL macro MNCL parameter) that may be entered on an Intercomm MNCL subsystem control command. Code as a decimal value, 1 to 32767. The default is 10. (SPAMMNCL)

MRAUTO

is a Multiregion parameter for satellite regions which specifies the action to be automatically taken by the satellite region if it detects that the control region is down. See Multiregion Support Facility. (SEXMRATO)

MRCNTL

specifies, for Multiregion systems, whether this is the Control Region. Code YES if this is the SPA for the Control Region. The default is NO. (SPAMRSW(SPAMRCON,YES=1,NO=0))

MRCSALN

is a Multiregion parameter for satellite regions under MVS and specifies, in bytes, the amount of CSA to acquire at startup time and hold until region termination. See Multiregion Support Facility. The default is 1024. (SEXMRCSL)

MRID

is a Multiregion parameter for satellite regions to specify the region identifier. See Multiregion Support Facility. (SEXMRID)

MR1LOG

is a Multiregion parameter for satellite regions to specify if this region's logging is to be done in the Control Region. (SPAMRSW(SPAMRLOG,YES=1,NO=0))

MSPR

specifies, in bytes, the minimum total amount of subpool 0 core that must be available before a subsystem can be forwarded any message. Code as a decimal value in multiples of 8, from 1024 to 512000. See NOTE under CUSHION parameter. The default is 2048. (SPAMSPR)

NQTIM

specifies, in seconds, the INTENQ macro TIME parameter default value; that is, the default time-out value within which an INTDEQ macro must be issued after the use of a resource has been granted. Code as a decimal value, 1 to 27962. The default is 60. (SPANQTIM)

NTIMS

specifies the maximum number of attempts that are to be made to obtain storage when a condition of no available dynamic storage exists, or to send output to a terminal when a persistent device busy condition is detected. Refer to the TIMS and DTIMS parameters. Code as a decimal value, 1 to 32767. The default is 7. (SPANTIMS)

NWAIT

specifies the maximum number of INTERCOMM WAITING messages to be printed when operating in Test mode. Intercomm terminates (see TSTEND parameter) after decrementing this value to a negative value. This parameter is meaningful only if WTO has been specified as an Intercomm EXEC parameter. Code as a decimal value, 0 to 32767. The default is 0. (SPANWAIT)

RCBSADD

specifies the number of new RCBs to acquire if all the RCBs currently allocated are in use. Code as a decimal value in the range of 0 to 32767 inclusive. Calculate as 2 to 5% of the value coded for RCBSINT. The default is 5. (SEXRCBUP)

RCBSINT

specifies the number of resource control blocks (RCBs) to be allocated and formatted at startup. This parameter is meaningful only if resource auditing has been selected; the Resource Management module (MANAGER) has been assembled with the global &RM set to 1. (See the Operating Reference Manual.) Code as a decimal value 0 to 32767 inclusive. Calculate as the total MNCL of all subsystems, plus the total number of files and terminals, plus 50. The default is 75. (SEXRCBLN)

RMSTIM

specifies the time interval, in seconds, between successive invocations of the detailed core use statistics calculation program. This parameter is meaningful only if the parameter COREACC is set to YES. The maximum value is 32767 (9 hours 6 minutes and 7 seconds). The default value is 5 seconds. (See also TRACETM parameter.) (SPARMST)

SEP

specifies the character to be used to indicate a message field separator for Intercomm system modules (Edit Utility, etc.). Translate the field separation character into EBCDIC and code as two hexadecimal digits. The default character is the comma (6B). The value coded must be identical to that coded for the SETENV symbolic variable &SEPCHAR. (See the Operating Reference Manual.) (SPASPCHR)

SGNTIME

specifies, in minutes, the default time interval to be used to automatically sign off a terminal after that terminal has been signed on; that is, the default duration a terminal may retain a signed on security clearance before that clearance is to be automatically revoked. This time interval will be used only when the following four conditions exist:

1. The sign-on/sign-off basic security system has been activated.

2. The terminal's STATION macro specifies AUTOFF=YES.

3. The terminal's STATION macro specifies TIME=0.

4. A sign-off transaction has not been requested. (The Intercomm SIGN system control command is used to sign on and sign off).

Code as a decimal value, 0 to 466. The default is 5 (minutes). A code of 0 signifies a default of no automatic sign-off. (SPASECTM)

NOTE: The sign-on/sign-off system is activated either at SPALIST assembly time via the SONOFF parameter, or during Intercomm execution time through the ASGN system control command. A terminal must be capable of receiving a security clearance. (Refer to the STATION macro, OPER parameter.)

SMCSWTO
lists ROUTCDE route codes (see PMIWTO and PMIWTOR macros) to be deleted from all PMIWTO and PMIWTOR executions. Unless 8 is given in this parameter, it is assumed to be part of the FMCSWTO parameter. (SEXSMCS)

SNAPPGS
specifies the number of pages of snap output to be accumulated before spinning off a data set via the SPINOFF facility to allow off-line printing (see the Operating Reference Manual). Code as a decimal value 0 to 32767. The default is 0, indicating that this facility is not desired (does not apply under MVS). (SEXSNAPP)

SONOFF
indicates whether or not the Intercomm basic security system is to be activated for possible use. Code YES if it is to be activated. Code NO if it is not be activated. The system control commands ASGN (activate sign-on/sign-off) and DSGN (deactivate sign-on/sign-off) provide the capability of setting and resetting this bit during execution time. The default is NO. (SPABITS2(SPASECNF,YES=1,NO=0))

SPMIWTO
lists ROUTX route codes (see PMIWTO and PMIWTOR macros) to be deleted from all PMIWTO and PMIWTOR executions (See also FPMIWTO parameter). (SEXSPMI).

SSNUM
defines the maximum number of functions controlled by the GPSS STRT/STOP system control commands. May be any value from 112 to 32752. The first 112 functions are reserved for Intercomm system use. The value is rounded up to a multiple of 8 for alignment. When the SPAEXT is assembled, an MNOTE is issued indicating the rounded value. The default value is 240. (BITSECT Csect).

STOCORE
specifies, in K, the total dynamic storage available to the Store/Fetch Facility for transient data strings kept in main storage at any given time on an Intercomm regionwide basis. (Refer to Store/Fetch Facility.) The default is 5 (5120 bytes). (SEXSFCSZ)

STRNO
specifies the maximum number of placeholders to be associated with the VSAM local shared resources pool. Code as a decimal value 1 to 255. The default is 1. (See also BLDVRP parameter.)

STSTIME

specifies, in seconds, the time interval betweeen issuances of system tuning statistics. (See the Operating Reference Manual.) Code as a decimal value in the range of 0 to 16777215. The default is 120 (2 minutes). (SEXSTSTM)

STUSPIE

specifies the program interruption types that are to be intercepted by the Intercomm SPIEEXIT. For maskable interruptions (fixed point overflow, decimal overflow, exponent underflow, significance), those specified are enabled and those not specified are disabled. Code in the format of the OS/VS SPIE macro. Default is ((1,13),15)--all maskable interruptions, except significance, are enabled. (SEXSPICA)

SWIN

specifies, in seconds, the duration of time allowed by the Subsystem Controller to elapse between the initiation of one scan to detect any work queued within any of the overlay region A Subsystem Control Table entries, and the initiation of the next such scan. This parameter is meaningful only if ECB=NO has been coded. Code as a decimal value, 1 to 27962. The default is 1. See also ECB and CNTDT parameters. (SPASWIN)

TASKNUM

specifies the number of general dynamic subtasks (1 to 210), followed by the number of user-coded special dynamic subtasks (0 to 15) in use. Intercomm requires at least one general subtask for its own use, but does not use special subtasks. Implementation requires the Intercomm module ICOMTASK in the linkedit. The default is (1,0). (SEXTASKC, SEXTASKS) (See also SUBTASK macro.)

NOTE: If the number of special dynamic subtasks is 0, the VCON for STASKTBL will be unresolved. This is not a problem if special subtasks are not to be used.

TCHP

specifies, in seconds, the interval of time between the execution of one checkpoint and the next. Code as a decimal value, 1 to 27692. The default is 120 (two minutes). (See also CKPTLIM parameter.) (SPATCHP)

TIMS

specifies, in seconds, the duration to wait after an attempt to acquire dynamic storage has failed and before another attempt is made. This value (multiplied by two minutes) also delimits the allowable elapsed time for the Output Utility to process a segmented message (from DVASN time to processing the X'53' message). (See the Utilities Users Guide.) Code as a decimal value, 1 to 27962. Refer to the DTIMS and NTIMS parameters. The default is 2. (SPATIMS)

TITECOR

indicates whether this is a small core system. YES indicates it is. The default is NO, indicating that conservation of storage is secondary to reduction of CPU usage. (SPABITS3(SPATCORE,YES=0,NO=1))

TOTATT

specifies whether or not the TOTAL data base management system is to be executed as a task separate from Intercomm within another region, or as an attached subtask under Intercomm within the same region. This parameter is for TOTAL data base system users only. Code YES if the system is to be attached. Code NO if the system is not to be attached. The default is YES. (SPABITS1(SPATOTRG,YES=0,NO=1))

TPMOD

designates the teleprocessing method employed. Code one of the following characters:

F -- BTAM - Intercomm BTAM Front End (code also if VTAM, GFE, and/or Extended TCAM Support is used) (default)
T -- Basic TCAM support only

A -- Intercomm BTAM (GFE) Front End and Basic TCAM in use (Extended TCAM not used).

(SPATPMOD(SPABFMOD,SPATCMOD))

TRACETM

specifies, in seconds, the interval between calls to RMTRACE, the routine that prints out core use statistics. This parameter has no effect unless the resource management module has been assembled with at least one of the globals &RMSTATS or &RMACCT set to 1. (See the Operating Reference Manual.) Code as a decimal value in the range 5 to 1800. The default is 120. (SEXTRINT)

TRANSEC

specifies whether or not the transaction basic security system on a terminal basis limiting terminal use is to be activated. Code YES if it is to be activated. Refer to the SECVERBS macro. The default is NO. (SPABITS2(SPASECVB,YES=1,NO=0))

TSTEND

specifies the method of termination for test mode runs:

DUMP--Abend 999 with dump (SPABITS3(SPATSTDP=1))
NODUMP--Abend 999 with no dump (SPABITS3(SPATSTDP=0))
NRCD--Generate Normal Closedown (default) (SPABITS3(SPATSTNR=1))

USERSEC

indicates whether or not user basic security routines are to be activated for possible use. The default is YES, indicating that the routines are to be activated. Code NO if they are not to be activated. Refer to the SYCTTBL macro SECU parameter. (SPABITS2(SPASECSC,YES=1,NO=0))

NOTE: If no user security routines are included in the linkedit and the default is taken, the servicing call instruction is bypassed.

WTO

specifies whether or not the message INTERCOMM WAITING is to be issued to the system console when an Intercomm 'no work' condition exists within overlay region A. Code NO if it is not to be issued. The WTO is also suppressed if WTO is not coded as an Intercomm EXEC parameter. The default is YES. (SPABITS1(SPANOWTO,YES=0,NO=1))

WTOPFX

provides the three-character prefix to be used as the Intercomm WTO identifier prefix on all messages generated via the PMIWTO(R) macro. (See also Messages and Codes.) The default prefix is INT. (SEXWTOPR)

SSSTART -- Start a STRT/STOP Function

This macro provides the ability to START a STRT/STOP function under program control. This is analogous in function to the STRT verb which may be entered at a terminal.

The form of the SSSTART macro is as follows:

[symbol]	SSSTART	[IF,] TYPE= {(r) / number / function} [,SPA=(r)][,SPAEXT=(r)]

The meanings of the operands are the same as the corresponding operands of the SSTEST macro.

SSSTOP -- STOP a STRT/STOP Function

This macro provides the ability to STOP a STRT/STOP function under program control. This is analogous in function to the STOP verb which may be entered at a terminal.

The form of the SSSTART macro is as follows:

[symbol]	SSSTOP	[IF,]TYPE={(r) / number / function} [,SPA=(r)] [,SPAEXT=(r)]

The meanings of the operands are the same as the corresponding operands of the SSTEST macro.

SSTEST -- Test a STRT/STOP Function

The SSTEST macro is provided to interrogate the status of STRT/STOP command-controlled functions. This permits external control of various functions (via the STRT/STOP verbs), while using the SSTEST macro internally to determine the current function status. The number of functions which may be so controlled is defined by the SSNUM operand of the SPALIST macro. The first 112 functions are reserved for Intercomm use.

If the number of functions to be controlled exceeds 240, the &SSMAX global in the SETGLOBE member must reflect this. SSNUM must never exceed SSMAX. At startup, all user and internal functions except for RMINTEG are set as started.

Following the execution of the SSTEST macro, the PSW condition code is set so that a BO instruction branches if the tested function is active (started) and a BZ instruction branches if the tested function is inactive (stopped).

The form of the SSTEST macro is:

```
[symbol]  SSTEST     parameters
```

The parameters are as follows:

`[IF,]`

IF is an optional positional parameter specifying that the presence of the STRT/STOP command capability is to be tested for. If not present, no interrogation is made; the program branches to the instruction following the SSTEST macro. If IF is omitted, interrogation is made unconditionally.

`TYPE={(r)}`
`{number}`
`{function}`

TYPE defines the STRT/STOP function to be tested. This may be defined in a register, with an absolute numeric value, or using certain reserved symbolic names. Symbolic names are valid only for Intercomm STRT/STOP functions. User functions must be referenced via a register or numeric value. User numeric values may range from 113 to the value of the SSNUM parameter of the SPALIST macro (and the value of &SSMAX in SETGLOBE if greater than 240).

(r) provides a register which contains a numeric value no greater than 32752.

function specifies the symbolic name of a STRT/STOP function administered by Intercomm. Valid specifications for function are:

Function	Meaning
LOG	writing to INTERLOG
RMSTAT	Printing Resource Management Statistics
RMINTEG	Resource Management Integrity Checking
FELOG	Front End Message Logging
BELOG	Back End Message Logging
FRLOG	File Recovery Logging
SAM	Statistics and accounting data gathering.
INDUMP	Indicative Dump option

`[,NO=address]`

NO specifies the branch address to be taken when the designated function is stopped. If NO is omitted, no branch is made.

If neither YES nor NO is specified, the user must code a branch instruction following the SSTEST macro in order to utilize the results of the test. If both YES and NO are specified, the program may not fall through to the instructions following the SSTEST macro.

`[,SPA=address]`

SPA specifies a register containing the address of the System Parameter Area. If this is specified the address of the STRT/STOP function control section is obtained from the SPA Extension. Otherwise, a literal V-type address is used.

(continued)

[,SPAEXT-(r)]
[,YES-address]

SPAEXT specifies a register containing the address of the SPAEXT. If this is specified the address of the STRT/STOP function control section is obtained from the SPA Extension. Otherwise, a lteral V-type address is used.

YES specifies the branch address to be taken when the designated function is started. It YES is omitted, no branch is made.

If neither YES nor NO is specified, the user must code a branch instruction following the SSTEST macro in order to utilize the results of the test. If both YES and NO are specified, the program may not fall through to the instructions following the SSTEST macro.

STALIST--Generate the Station Table DSECT

The STALIST macro is used to generate the DSECT for the Station Table (header plus one entry).

The STALIST macro has no parameters.

The form of the STALIST macro is as follows:

```
(blank)     STALIST
```

STATION -- Specify Terminal Information

The STATION macro specifies various elements of information required about terminals by the Intercomm Back End. The STATION macro creates the PMISTATB table entries that furnish ESS and the Output and MMU utilities with required information. Also, in conjunction with the GENSEC and SECVERBS macros, the STATION macro creates the PMISECTB table entries for the Transaction and Sign-on/Sign-off features of Basic Security. The last STATION macro must be followed by a PMISTOP macro.

The form of the STATION macro is as follows:

(symbol)	STATION	terminal parameters: TERM=terminal-id ,IOCODE={dir-and-devtype-code } {(dir,devtype-name(,dvmodify-label))} (,ALT=alternate-terminal-id) {,MASK={output-hex-mask}} { {FFFF } } {,USE={DOWN}} { {UP } } sign-on/sign-off security parameters: {,AUTOFF={NO }} { {YES}} {,MAXSIGN={max-num-sign-on-attempts}} { {3 } } {,OPER={sign-on-code }} { {(code(,...,code))}} {,RBN={SEC000-rbn}} { {0 } } {,TIME={sign-off-time-interval}} { {0 } }

(continued)

		transaction-id security parameter: [,VERBS={security-transaction-id }] [{(trans-id[,...,trans-id])}] remote job entry parameters: [,RJECNTL={YES}] [{NO }] [,RJE={YES}] [{NO }] macro type: [UNIVER={YES}] [{NO }] continuation parameter: [,CONTIN={YES}] [{NO }]

symbol
: is any symbol valid in Assembler Language and may be up to seven characters in length. This symbol must be coded on the first nonuniversal STATION macro; internally generated labels are developed from it. No symbol may appear on subsequent STATION macros.

ALT
: specifies the Intercomm terminal-id of a terminal to be used as an alternate if the subject terminal is down. The alternate device type must be the same as that of the subject terminal. (See also USE parameter.)

AUTOFF
: specifies whether or not the terminal is to use the automatic sign-off feature of the sign-on/sign-off Basic Security system. This parameter is valid only if security codes have been assigned to the OPER parameter. Code NO if this feature is not to be used. The default is YES. The automatic sign-off duration interval must be provided via either the TIME parameter or the SPALIST macro, SGNTIME parameter.

CONTIN

specifies that the information about this terminal is being continued on a succeeding STATION macro. This is necessary if the OPER or VERB parameters would otherwise exceed 255 characters (including commas and parentheses), the maximum for a macro parameter, or if more than 50 operator codes or verbs must be defined for this terminal. Code CONTIN=YES if this STATION macro is being followed by another STATION macro with information on the same terminal. Do not repeat parameters on continuation macros, except for those specifying additional values. The default is NO. A maximum of 512 verbs or operator codes may thus be defined for one terminal.

IOCODE

specifies input/output type and device type. If UNIVER=NO, this parameter is required unless CONTIN=YES was coded on the preceding STATION macro. There are two ways to code this parameter, by device type number or by device type name. Care must be exercised in choosing the appropriate method. Both are valid for Output Utility format processing, but only the latter method (by device type name) can be used for MMU format processing.

If IOCODE is specified by device type number, device type numbers are restricted to the maximum of 16 which can be defined via the DEVICE macro. IOCODE is specified as two hexadecimal digits. The first digit specifies permitted direction of transmission (1 for input, 2 for output and 3 for both). The second digit specifies the user-assigned code corresponding to a device type code defined via a DEVICE macro TYPE parameter.

If IOCODE is specified by device type name, there is no restriction on the number of devices which can be defined. Also, it can support more varied forms of the same device types or compatible devices.

dir is the code for the permissible direction of transmission. Code 1 for input; 2 for output; 3 for input and output.

devtype-name is the generic name of the device type, and must be the same as that defined for the corresponding DEVICE macro, TYPE parameter. The permissible values are:

devtype-name	Device Type
DS40	Teletype Dataspeed 40/1 and 2
IBM27401	IBM 2740--Model 1
IBM27402	IBM 2740--Model 2
IBM3270	IBM 3270 CRT
IBM3270P	IBM 328x Printer (3270 series)
IBM2741	IBM 2741
IBM2770	IBM 2770
IBM2780	IBM 2780
IBM2260	IBM 2260
IBM1053	IBM 1053
IBM1050	IBM 1050
IBM1030	IBM 1030
IBM360	IBM 360/370 (remote CPU)
IBM3735	IBM 3735
IBM7770	IBM 7770
TELETYPE	Teletype (and compatible) Devices
ATT83B3	AT&T 83B3
SAND620	Sanders 620
SAND720	Sanders 720
BKRAMO	Bunker Ramo 2200
U100	Uniscope 100
ULT7700	Ultronics 7700

dvmodify-label is the name of the DVMODIFY macro which further describes this device, if defined.

MASK

specifies the classes of Output Utility reports the terminal is to be considered eligible to receive. This mask is to be coded with respect to the report classification relation noted under the REPORT macro MASK parameter in the Utilities Users Guide. The code is to be thought of as a 32-bit string written as a one- to four-character hexadecimal string with all codes not four characters being first right-adjusted, then left-padded with zeros. FFFF, the default, specifies that the terminal is eligible to receive all reports from all reporting format classes.

MAXSIGN

specifies the maximum number of times an operator can enter the SIGN command and fail in the attempt to pass sign-on/sign-off security for the terminal. Failure to sign-on is recorded at the Control Terminal; failure to sign-on within the specified number of attempts results in the immediate cessation of polling of the terminal with notification again sent to the Control Terminal. A TPUP command is then required to place the terminal on-line again. Code as a decimal value, 1 to 255. The default is 3.

OPER

specifies the one or more operator codes permitted to be entered at the terminal. If OPER is not coded, the sign-on/sign-off Basic Security feature is not required. If more than one security code is to be specified, they must be coded as a parameter sublist. If UNIVER=YES, OPER must be coded. Each operator code must be in decimal numeric form, one to ten characters in length, and not outside the value range 1 to 2147483647. The maximum number of codes that can be defined on one STATION macro is 50 (see CONTIN and RBN parameters). Refer also to the GENSEC macro.

RBN

provides the relative block number of the SEC000 file at which the subject terminal's security codes can be located. If the associated security codes are to be core-resident, this parameter is not valid. However, if the codes are to be located within the SEC000 file, the RBN value is to reflect the last five digits of the member-name used to place the entry in the file. For example, if the member-name is SEC00008, then code RBN=8. Code as a decimal value, 1 to 32767. Refer to the GENSEC macro.

RJE

indicates whether or not the terminal is permitted to submit RJE input under OS/MFT or OS/MVT only. (This parameter applies to RJE input only; RJE output may be sent to any terminal via a remote command.) The default is NO unless RJECNTL=YES, in which case the default is YES.

RJECNTL

specifies whether or not the terminal is permitted to submit RJE control commands for terminals other than itself. Code YES if the terminal is permitted to submit such commands. The default is NO. (Requires RJE Facility: OS/MFT and OS/MVT only.)

TERM

specifies the five-character Intercomm terminal-id to be associated with the terminal and which corresponds to a like-named terminal defined in the Intercomm Front End Network Table. If UNIVER=NO, this parameter is required unless CONTIN=YES was coded on the preceding STATION macro.

TIME

specifies, in minutes, the duration the terminal may retain a signed-on Basic Security clearance before that clearance is to be automatically revoked. This parameter is valid only if AUTOFF=YES has been specified. A zero indicates that the interval specified by the SPALIST macro SGNTIME parameter is not to be overridden. Code as a decimal value, 0 to 466. The default is 0.

UNIVER

specifies whether or not this STATION macro is a universal instruction supplying a list of operator codes valid under Sign-on/Sign-off Basic Security for all the terminals in the network. There can be only one STATION macro for which UNIVER=YES, and it must precede all other STATION macros. If UNIVER=YES, the only other parameter to be coded is OPER. The default is NO.

USE

indicates, at startup time, whether the terminal is to be considered on-line or off-line. Code DOWN if the terminal is to be considered off-line. The default is UP, indicating that the terminal is to be considered on-line (recommended).

VERBS

indicates whether or not the transaction-id Basic Security feature is required at this terminal. If VERBS is not coded, transaction security is not required. The presence of the VERBS parameter indicates that it is required and specifies the secured transaction-ids permitted to be entered at the terminal. If more than one such transaction-id is to be specified, they must be coded as a parameter sublist. The maximum number of verbs that can be defined on one STATION macro is 50 (see CONTIN and RBN parameters). Refer also to the GENSEC and SECVERBS macros.

If transaction-id security is desired, and the VERBS parameter is not coded on this STATION macro, all secured verbs may be entered via this terminal. To prevent the terminal from entering any secured verbs, code a dummy verb (any four characters) in the SECVERBS macro and code the VERBS parameter naming the dummy verb only.

STORAGE -- Obtain Main Storage

The STORAGE macro instruction issues a request for ownership of a portion of core. If the request is satisfied, register 15 will contain a return code of 0 and the core allotted will commence on a double-word boundary. If the request is not satisfied, register 15 will contain a return code of 8. The ownership of core areas obtained via the use of the STORAGE macro instruction must be relinquished through the use of the STORFREE macro instruction.

The form of the STORAGE macro instruction is as follows:

[symbol]	STORAGE	[ADDR={address / (r)}] ,LEN={length / (r)} [,SPA=(r)] [,SP={subpool / 0 / (r)}] [,ZERO={YES / NO}] [,LIST={list-address / (r) / (1)}] [,SYS={YES / NO}] [,RENT={YES / NO}] [,CONDL={YES / NO}] [,POOLS={YES / NO}] [,BOUND={DOUBLE / PAGE}] [,ERRADDR={(r) / address}]

ADDR=
provides the address of a full-word where the address of obtained storage may be returned. If this operand is omitted, the address of the obtained storage is returned in register 1.

LEN=
provides the length, in bytes, of main storage to be requested. "Length" may be any self-defining or arithmetic expression or may be specified in a register. Note that if the length value is not a multiple of 8, it will be rounded upwards to the next higher multiple of 8. Note also that the value may not exceed the length of core available to INTERCOMM. This operand is required.

SPA=
designates the general register (2-12) that contains the address of SPA. If this is omitted, a literal V-type address constant will be generated to obtain the STORAGE routine's entry point.

SP=
specifies the subpool number. Code as a decimal value, 0 to 127, or specify in a register. The default code is 0.

ZERO=
indicates whether or not the area of core obtained is to be zeroed. Code NO if the area is not to be zeroed. Code YES if the area is to be zeroed. The default code is YES.

LIST=
provides the address of a three-word area available for use. "Address" may be any expression that can be substituted into the second operand field of an LA instruction or may be specified in a register. If this parameter is omitted, then "RENT=NO" will be forced.

SYS=
in a system including Resource Auditing and Purging, this parameter is used to assign ownership of the storage area to INTERCOMM, ensuring that it will never be purged. Code YES to assign the area to INTERCOMM. The default code is NO.

RENT

indicates whether Resource Management should wait and retry if the storage request cannot be filled. The default, RENT=YES, means retry; register 13 must be pointing to a valid savearea and the LIST parameter must be coded. If LIST is omitted, RENT=NO will be forced. RENT=NO tells Resource Management to suppress retries if storage is not available; in this case it is not necessary to supply either a register save area or space for a list; thus a STORAGE macro with RENT=NO may be used to get a save area.

CONDL

specifies whether the STORAGE request is conditional or unconditional. If CONDL=YES is specified, then Resource Management will return a code of 8 in register 15 if the request cannot be satisfied. If CONDL=NO is specified, and the request cannot be satisfied, an intentional OC2 (via ISK) program check will occur, terminating the thread, and message RM023I will be issued. The default is YES.

POOLS

specifies whether an attempt should be made to satisfy the request from Intercomm's Resource Management pools. If POOLS=YES is coded, an attempt will be made. If POOLS=NO is coded, then a GETMAIN for an OS/VS subpool area will be forced. The default is YES.

BOUND

specifies the boundary alignment of the storage request. This operand is meaningful only in a VS environment. BOUND=DOUBLE specifies doubleword alignment. BOUND=PAGE specifies that the storage area is to be on a virtual page boundary. BOUND=PAGE is ignored when not running under VS. BOUND=DOUBLE is the default.

ERRADDR

specifies the address of an error routine to be given control if storage cannot be obtained. A symbolic address or an address in a register may be provided. An accompanying specification of CONDL=NO will be ignored.

STORFREE -- Free Main Storage

The STORFREE macro instruction frees an area of main storage previously acquired through the use of a STORAGE macro instruction.

The form of the STORFREE macro instruction is as follows:

[symbol]	STORFREE	LEN={length / (r) / (0)},ADDR={address / (r) / (1)} [,SP={subpool / 0 / (r)}] [,LINK=(15)] [,SPA=(r)]

LEN=

specifies the length, in bytes, of main storage to be freed. "Length" may be any expression that can be substituted into the second operand field of an LA instruction (e.g., 208,EQUATE,0(2,4)) or may be specified in a register. Note that if the length value is not a multiple of 8, it will be rounded upwards to the next higher multiple of 8.

ADDR=

provides the address within main storage from which the requested number of consecutive bytes are to be freed. Unlike the STORAGE macro ADDR= parameter, this is an actual address and is not a fullword containing the address. "Address" may be any expression that can be substituted into the second operand field of an LA instruction or may be specified in a register. Note that this address must be on a doubleword boundary.

SP=

specifies the subpool number. Code as a decimal value, 0 to 127, or specify in a register. The default code is 0.

LINK=(15)

indicates that the STORFREE entry point has been previously loaded into register 15. If used, this parameter must be coded exactly as shown.

SPA=

designates a general register that contains the address of the SPA. If this is omitted a literal V-type address constant is generated to obtain the entry point of the STORFREE routine. If LINK is specified, this operand is ignored.

SUBLINK--Provide Subroutine Linkage

The SUBLINK macro generates the linkage code required by a reentrant Assembler Language subroutine (system or user) or subroutinized (dispatched) code, and should be paired with a RTNLINK macro at exit from the subroutine. This macro should always be used for dynamically loadable and Link Pack eligible subroutines. The macro performs several functions:

- Acquire and chain a save/work area
- Zero the acquired area
- Save caller's registers
- Initialize base register(s)
- Provide parameter area addressability
- Provide SPA/SPAEXT addressability

The form of the SUBLINK macro is as follows:

[symbol]	SUBLINK	LEN=save/work-area-length [,BASE={(r) }] [{(r1,r2)}] [,GEN={NO }] [{YES}] [,PARM=(r)] [,SAVE={NO }] [{YES}] [,SPA={(r) }] [{label}] [{(1) }] [,SPAEXT={(r) }] [{label}] [{(0) }] [,ZERO={NO }] [{YES}]

symbol

may be any label valid in Assembler Language and up to eight characters in length. It will be the label of a SAVE macro if SAVE=YES, or of a DS 0H to provide alignment.

BASE

specifies the single register or a pair of registers to be initialized as the base register(s) of the routine (USING also generated) and assumes register 15 contains the address of this SUBLINK macro. If omitted, this macro must be coded within the range of a predefined base register and USING statement. If coded, SAVE=NO may not be coded.

GEN

specifies whether or not the PMISUBL2 CSECT called by this macro should be generated in-line. If NO is coded, this subroutine must be linkedited with another subroutine for which the CSECT has been generated (unless this subroutine is dynamically loadable and dynamic linkedit is used). The default is YES, in which case the CSECT will be generated unless already generated by an earlier SUBLINK macro in the same subroutine.

LEN

specifies the length, in bytes, of the save/work area to be acquired, and is a required parameter. The length may be a decimal value, a self-defining term (equated symbol), or may be provided in a general register (2-12) or in register 0. The length will be automatically rounded up to a doubleword value. Save area chaining will be performed and register 13 will point to the acquired area which the user must free at exit (via RTNLINK or STORFREE macros).

PARM

designates the general register (2-12) that is to contain the address of the parameter list (if any) passed via register 1. The program must provide and establish DSECT addressability (if desired).

SAVE

specifies whether or not the first instruction of the macro expansion should be a SAVE macro (to save caller's registers). If NO is coded, the user has that responsibility. The default is YES.

SPA

designates the general register (2-12) or field (in the save/work area) to hold the address of the SPA Csect (obtained internally via GETSPA macro). If SPA=label is coded, label must be that of a fullword predefined field (unless defined via the Dsect for the work area - user must establish addressability via precoded USING statement). If omitted, the SPA address is returned in register 1. A USING must be coded and the SPALIST Dsect must be defined (if desired).

SPAEXT

designates the general register (2-12) or field (in the save/work area) to hold the address of the SPAEXT Csect (obtained internally from the SPA). If SPAEXT=label is coded, it must reference a predefined field (see SPA parameter). If omitted, the SPAEXT address is returned in register 0. A USING for the SPAEXT portion of the SPALIST Dsect must also be coded (if desired).

ZERO

specifies whether or not the portion beyond the first 72 bytes of the acquired save/work area should be initialized to binary zeros. The default is YES.

NOTE: The SAVCELL parameter may be used only by Intercomm Link Pack eligible system routines, and is not documented here.

SUBMODS -- Define User-Coded Subroutine

The SUBMODS macro is coded in the REENTSBS table, or the user-coded COPY member USRSUBS, to define user-coded subroutines which may be resident or dynamically loaded. Subsystems which use COBREENT or PMIPL1 may access these subroutines in the same manner as resident Intercomm routines. This macro must also be coded for BAL subsystems accessing dynamically loaded subroutines (see MODCNTRL macro).

The form of the SUBMODS macro is as follows:

[symbol]	SUBMODS	{NAME=module-name } {LNAME=module-name} [,BLDL={NO }] [{YES}] [,DELTIME={deletion-interval}] [{0 }] [,GET={nnn}] [{0 }] [,PARM={(MSG,SPA,SCT,RC)}] [{ALL }] [{NO }] [,PERMRES={YES}] [{NO }] [,RES={LOADMOD }] [{LINKEDIT}] [{BOTH }] [,TYPE={COBOL}] [{PL1 }] [{BAL }] [,USAGE={REUSE }] [{NONREUSE}] [{REENT }]

(continued)

	CICS/CF Parameters: [,CICSCF={YES}] [{NO }] [,SUBC={lo-order-entry code}] [{000 }] [,SUBH={hi-order-entry-code}] [{000 }] [,TRANID=transaction-id]

BLDL

specifies whether or not a BLDL-list should be maintained for a dynamically loadable subroutine. BLDL-lists reduce the overhead involved in loading a subroutine, but require an extra 60 bytes of storage per subroutine. The default is YES.

CICSCF

specifies whether or not the CICS Compatibility Feature is in use with this subroutine. The default is NO. See CP manual.

DELTIME

specifies the time interval, in seconds, of subroutine inactivity which must elapse before the loaded subroutine is deleted from core. Code as a value from 0 to 63. Specifying a nonzero interval allows actively-used subroutines to be reused repeatedly (subject to the USAGE parameter) without being reloaded. The default is 0, which causes the subroutine to be deleted as soon as it becomes inactive.

GET

defines, in bytes, the amount of Dynamic Working Storage to acquire and pass (ahead of the caller's parameter list) to a reentrant COBOL subroutine (in the same manner as DWS is passed to a COBOL subsystem). Code as a numeric value in the range 8 to 131072 (128K). If coded, TYPE=COBOL and USAGE=REENT (default) are also required. The default is 0 (do not acquire DWS).

LNAME

specifies the entry point name of the subroutine if it is eligible for dynamic subroutine loading and/or is not written in reentrant BAL. This parameter is mutually exclusive with the NAME parameter. See RES, TYPE and USAGE parameters.

NAME

specifies the entry point name of a resident subroutine. This parameter may be used only if the subroutine is written in reentrant BAL and is resident in the linkedit. NAME is mutually exclusive with LNAME. No other parameters may be coded.

PARM

specifies the system parameters (input message, SPA, SCT, return code) to be passed to a reentrant COBOL subroutine before the DWS (if GET is coded) and caller's parameter list. Some, all or no system parameters may be requested. If some parameters are desired, code as a sublist (in parentheses and separated by commas) one or more of the values MSG, SPA, SCT and/or RC. ALL requests that all four parameters (in the above order) be passed. Do not code ALL or the MSG subparameter if the input message was freed by the caller (via a call to MAPIN, for example). NO (default) requests that no added parameters be passed because they are passed by the caller, or are not desired.

PERMRES

specifies, for a dynamically loadable subroutine, whether it should be loaded at startup and made permanently resident (recommended if the subroutine performs file I/O or enqueue/dequeue requests). If YES, it may not be deleted, nor reloaded via the LOAD command. The default is NO.

RES

specifies whether the subroutine is a separate load module (LOADMOD) or is resident in the main linkedit (LINKEDIT). The default is BOTH, specifying that provision is to be made to dynamically determine where the subroutine resides.

SUBC

is used with the CICS Compatibility Feature.

SUBH

is used with the CICS Compatibility Feature.

TRANID

is used with the CICS Compatibility Feature.

TYPE

specifies the language employed by the subroutine. The default is BAL. If PL1 is specified, invocation of the PMIPL1 interface allows a PL1 format parameter list (locator/descriptors) to be passed to the subroutine.

USAGE

specifies whether the subroutine is coded as reentrant, reusable or nonreusable. The default is REENT (required if the subroutine calls COBREENT or PMIPL1, or, if BAL, if it gives up control directly or indirectly via the dispatcher).

SUBTASK -- Execute a Routine as a Subtask

The SUBTASK macro allows part of a thread's logic to execute as a subtask. The program-linkage between the main Intercomm task and the subtasked logic may be viewed as being equivalent to a CALL to a subroutine; the registers of the thread issuing the SUBTASK macro will be preserved if restored by the subtasked routine before exit, and modified if altered by the subtasked routine, but not restored. Registers 1, 13, 14 and 15 can, therefore, be used as if a CALL was issued. (SUBTASK generates the instruction BALR 14,15 with register 15 containing the ENTRY parameter value, as described in the Operating Reference Manual.)

The code executed as the subtask cannot use Intercomm service functions; Dispatcher, File Handler, etc. Execution of the subtask logic is synchronous with respect to the thread issuing the SUBTASK macro.

The form of the SUBTASK macro is as follows:

[symbol]	SUBTASK	ENTRY={address} {(r) } [,PARM={address}] [{(r) }] [{(1) }] [,{SPA }=(r)] [{SPAEXT}] [,TASKNUM=hex-digit]

ENTRY
: specifies the address of the entry point of the logic to be subtasked.

PARM
: specifies the parameter list address forwarded to the subtask. The default is register 1.

SPA
: specifies the general register containing the address of the SPA. If this parameter is coded, the address of the Intercomm ICOMTASK processing routine will be obtained from the SPA Extension. Otherwise, a V-type address constant will be generated.

SPAEXT
: specifies the general register containing the address of the SPAEXT. If this parameter is coded, the address of the Intercomm ICOMTASK processing routine will be obtained from the SPA Extension. Otherwise, a V-type address constant will be generated.

TASKNUM=

requests the execution of a special subtask. The value coded must be a valid special subtask-ID within the range coded on the SPALIST macro TASKNUM special subtask parameter value. Code as a hexadecimal digit from 1 to F. If TASKNUM is not coded, the subroutine will execute under a general subtask.

SYCTTBL -- Define Subsystem

The SYCTTBL macro is used to define a subsystem to Intercomm. It defines an entry within the Subsystem Control Table (in module INTSCT or the user-coded COPY member USRSCTS) and provides the identification, queuing and procedural specifications required to control the subsystem. A maximum of 9999 SYCTTBL macros may be coded. See the Operating Reference manual.

Whether a SYCTTBL macro represents one subsystem or multiple subsystems depends on how the subsystems are to be linkedited into the system. For each subsystem that is to be either core-resident, dynamically loadable, or residing in overlay Region A, there must be one associated SYCTTBL instruction. However, for one or more subsystems that are to reside in each of the overlay Regions B, C or D, there can be only one corresponding region-associated SYCTTBL instruction. The distinction between subsystem-associated SYCTTBL instructions and region-associated SYCTTBL instructions is maintained throughout the parameter specifications.

In addition, the SYCTTBL macro is also used to generate the queue table entries for Intercomm's BTAM Front End (BTAMSCTS). These entries control queues of messages waiting for transmission to terminals. For the VTAM Front End, SYCTTBL specifications (VTAMSCTS) are generated internally from parameter coding on the LUNIT/LCOMP macros.

The form of the SYCTTBL macro is as follows:

[symbol]	SYCTTBL	macro-type parameter: [TYPE={BTAM}] [{INT }] identity parameters: [,LANG={NBAL}] [{RBAL}] [{COB }] [{RCOB}] [{PL1 }] [{RPL1}] [{FORT}] [,OVLY={overlay-code}] [{0 }] [,SBSP={subsys-entry-point}] [{MONOVLY }]

(continued)

```
                    [,SUBC=low-order-entry-code]

                    {,SUBH={high-order-entry-code}}
                    {      {000                  }}

                    basic-TCAM queue parameter:

                    [,DCBP=process-q-dcb-pointer]

                    core queue specifications:

                    {,NUMCL={number-core-q-entries}}
                    {       {0                    }}

                    {,PRYMSGS={number-priority-q-entries}}
                    {         {0                        }}

                    disk queue specifications:

                    {,BLRI={B }}
                    {      {N }}
                    {      {NQ}}
                    {      {F }}

                    [,DFLN=disk-queue-ddname]

                    {,PCEN={units             }}
                    {      {(units,hundredths)}}
                    {      {100               }}

                    execution parameters:

                    {,ECB={NO }}
                    {     {YES}}

                    [,GET=core-amount[,FREE=core-amount]]

                    {,MNCL={max-concurrent-executions}}
                    {      {1                        }}

                    {,PRTY={priority-code}}
                    {      {0            }}

                    {,SCHED={NO }}
                    {       {YES}}

                    {,SPAC={req-amount-subpool-0-core}}
                    {      {5000                     }}
```

(continued)

		(,TCTV={subsys-execution-time}) ({60 }) (,THRSH={scheduling-threshold-num}) ({0 }) dynamic subsystem load parameters: (,BLDL={NO }) ({YES}) (,LOADNAM=load-module-name) (,REUSE={NO }) ({YES}) VS execution parameters: (,EXGRP=execution-group-number) security parameters: (,SECU={security-routine-index}) ({0 }) (,SOSO={YES}) ({NO }) (,TISE={YES}) ({NO }) procedural parameters: (,AUXS=auxiliary-queue-pointer) (,BACKOUT={NO }) ({YES}) (,CANC={STOP }) ({CONTINUE}) (,CNVREST={YES}) ({NO })

(continued)

```
[,DBASE={ADA    }]
        {UADA   }
        {DL1    }
        {UDL1   }
        {DB     }
        {UDB    }
        {TOTAL  }
        {UTOTAL }

[,IMCDFL={YES}]
         {NO }

[,LOG={NO }]
      {YES}

[,LSYNCH={YES}]
         {NO }

[,PL1={OPT}]
      {F  }

[,PL1LNK={BASED    }]
         {NONBASED }

[,RESOURC=identifier]

[,RESTART={IFPOSBL}]
          {NO     }
          {YES    }

[,RVFILE={YES}]
         {NO }

[,SEGREST={YES}]
          {NO }

[,SNAP={NO }]
       {YES}

[,WTO={NO }]
      {YES}

CICS/CF parameters:

[,CFTWA={size}]
        {0   }

[,CICSCF={YES}]
         {NO }
```

(continued)

		VTAM parameter: [,SRESP={NOSPEC }] [{{D},{2}}] [{{E} {1}}]

In the following parameter specifications, one or more capitalized symbols in parentheses refer to labels within the SCTLISTC DSECT directly associated with the parameter. For example, an expression such as

(SCTBITS(SCTMIN,YES=1,NO=0))

means that if NO is coded on the associated parameter, the bit identified by the equate label SCTMIN, and residing within the field identified by the label SCTBITS, is set to 0. One or more symbols parenthetically qualifying another imply relevant equated bits within a field.

AUXS

specifies the label identifying the SYCTTBL macro defining the entry from whose queues work is to be extracted if the subject subsystem or region:

- does not possess any queues,
- possesses a complete absence of work within its queues, or
- possesses an insufficient amount of work within its queues in terms of the capacity of its own MNCL value.

If a subsystem or region-associated instruction does not specify either a core or disk queue, this parameter must be coded.

This parameter provides the ability to have multiple table entries point to a single queue; however, a table entry whose associated queues are so made accessible to other entries may not itself designate an alternate work source queue. (SCTAUXSS(displacement from auxiliary to SCXESCX))

BACKOUT

specifies whether or not the subsystem is eligible for Backout-on-the-Fly (if implemented in the system - see File Recovery Users Guide). The default is YES. (SCTBIT1(SCTBKOUT,YES=1,NO=0))

BLDL

indicates, for a dynamically loaded subsystem, whether a BLDL list (which occupies 60 bytes of storage per directory entry) should be maintained for the load module. A BLDL list allows a module to be loaded more rapidly than otherwise. This parameter is only valid if the LOADNAM parameter is specified. Code NO if the BLDL list is not to be maintained. If REUSE=NO, BLDL=NO must be coded. The default is YES, indicating that the list is to be maintained. (SCTBIT3(SCTBLDL,YES=1,NO=0)).

BLRI

specifies the kind of message queuing for the subsystem or region. This parameter is valid only if the DFLN parameter is coded. Message queuing is specified as follows:

B -- a core and a disk queue are both being used; one or more messages may constitute a single disk queue block; and the order of messages input need not determine the order of messages forwarded to the subsystem or region (that is, FIFO order is not required). A received message is granted immediate core queue priority, the disk queue functions merely as an overflow facility. The core and disk queues are each processed independently in FIFO order, and so a message received before another may be processed after it. (SCTBIT2(SCTBLK))

F -- (default) a core and a disk queue are both being used; one or more messages may constitute a single disk queue block; and the order of messages input determines the order of messages forwarded to the subsystem or region (that is, FIFO order is required). This code must be used if segmented input is to be received by a subsystem, and is required for a terminal queue. A received message is directed immediately to the end of the current queue; the disk queue is treated as an expansion of the core queue. The entire queue is processed in FIFO order. (SCTBIT2(SCTFIFO))

N -- a disk queue is being used and a core queue may or may not also be in use; and only a single message may constitute a single disk queue block. If a core queue is used, the integrity of the order of messages input cannot be kept; if not used, input order integrity is maintained. (SCTBIT2(SCTNOBLK))

NQ-- only a disk queue is being used; one or more messages may constitute a single disk queue block; and that the order of messages input need not determine the order of messages forwarded to the subsystem or region (that is, FIFO order is not required). (SCTBIT2(SCTBNCQ)).

CANC

specifies the subsequent action to be taken by the Subsystem Controller after the first occurrence of either a subsystem time-out (see the TCTV parameter), or a subsystem return code of 8, or a subsystem program check. Code STOP if the subsystem (or region) is not to receive any further work. If STOP is coded, a user-supplied routine possessing a CSECT name of USRCANC must be written to handle the balance of messages (see released member PMICANC). The default is CONTINUE, specifying that the subsystem (or region) is to receive further work. (SCTBITS(SCTSTOP,STOP=1,CONTINUE=0)).

CFTWA

for CICS/CF (CP) subsystems, specifies, in bytes, the length of the USRTWA. Code in decimal in the range 2 to 32766. The default value is 0. This parameter is valid only if CICSCF=YES. (SCTTWA).

CICSCF

specifies whether or not this is a CICS macro-level application program to be run under the CICS Compatibility Feature (CP). Code YES if it is. The default is NO. (SCTTWA+1-bit7=1).

CNVREST

specifies how message restart is to handle the case where the subsystem was processing a message received via CONVERSE when a failure occurred. YES causes the first message in the conversation to be restarted; all the messages the subsystem received via CONVERSE are discarded, complete or not. NO (default) causes the message being processed to be restarted in the normal way. This parameter is valid only if RESTART=YES or RESTART=IFPOSBL. (SCTBIT2(SCTCVRST,YES=1,NO=0)).

DBASE

subsystem-associated instruction

specifies the use of a data base management system by the subsystem.

region-associated instruction:

specifies the use of a data base management system by one or more subsystems within the region.

The data base utilization is specified as follows:

CODE	DATABASE	ACCESS ONLY	ACCESS AND UPDATE
ADA	ADABAS	X	
UADA	ADABAS		X
DL1	DL/1	X	
UDL1	DL/1		X
DB	Generalized Data Base	X	
UDB	Generalized Data Base		X
TOTAL	TOTAL	X	
UTOTAL	TOTAL		X

(SCTBIT1(SCTDB,SCTDBUPD))

DCBP

specifies the label identifying the address of the TCAM process queue DCB for the subsystem if Basic TCAM Support is the teleprocessing interface. This is not used for Extended TCAM Support via GFE. (SCTDCBP)

DFLN

specifies the ddname of the DD statement defining the data set, all or part of which is used as a terminal's, the subsystem's, or region's disk queue. Refer to the PCEN parameter. (SCTDFLN)

ECB

subsystem-associated instruction:
specifies whether or not the presence of work destined for a dynamically loaded or a core resident subsystem is to be indicated by the posting of the entry's ECB (SCTRECB). For subsystems residing in overlay region A, refer to the SPALIST macro ECB parameter.

region-associated instruction:
specifies whether or not the presence of work destined for a subsystem within the region is to be indicated by the posting of the entry's ECB.

Code NO if the ECB is not to be posted and work is to be detected via a regularly dispatched search for work. Refer to the SPALIST macro SWIN and CNTDT parameters for definable workscan intervals. The default is YES, indicating that the ECB is to be posted. (SCTBIT1(SCTECB,YES=1,NO=0))

EXGRP

defines VS Execution groups. An execution group is a collection of one or more resident subsystems that have been linkedited into a common set of VS pages via the ORDER statement. This option limits the scheduling of subsystem execution in a fashion similar to that of Overlay Region A scheduling, and is intended to replace the use of the overlay supervisor with the VS paging supervisor. If coded, it overrides the OVLY parameter. Code as a numeric value between 4 and 62. (SCTPONU, bits 2-7)

FREE

indicates, in bytes, the amount of the DWS provided on entry (via GET) to a reentrant COBOL subsystem, which should be freed when the subsystem completes. This parameter is only valid if LANG=RCOB and the GET parameter is also specified. It defaults to the value specified for the GET parameter. Code as a decimal value which is a multiple of eight. DWS checking, if used, requires that FREE be omitted. (DYNREQ1 CSECT entry-bytes 5-6)

GET

specifies, in bytes, the amount of dynamic working storage to be provided on entry to a reentrant COBOL subsystem. This parameter is valid only if LANG=RCOB. Code as a decimal value which is a multiple of eight up to 65232 bytes (do not include storage for DWS checking). (DYNREQ1 CSECT entry - bytes 2-3)

IMCDFL

specifies whether or not, when an IMCD (immediate closedown system command) is issued, the subsystem or region's input queues are to be flushed (that is, its messages processed) before the system itself is halted. Code YES if the subsystem's queues are to be flushed. The default is NO. (SCTBIT1(SCTIMCFL,YES=1,NO=0))

LANG

subsystem-associated instruction:

specifies the programming language the subsystem is written in, and whether or not the subsystem is reentrant. If a subsystem is to receive segmented input, it must be treated as non-reentrant, either via MNCL=1 or via the LANG parameter.

region-associated instruction:

specifies the programming language all subsystems or subsystem interface programs in the region are written in. All associated subsystems are treated as non-reentrant whether they are coded as such or not, even if the region is defined as reentrant.

(SCTLANG(SCTNBAL,SCTRBAL,SCTCOB,SCTRCOB,SCTPL1,SCTRPL1,SCTFORT))

The language is specified as follows:

NBAL - Nonreentrant Assembler Language
RBAL - Reentrant Assembler Language
COB - Nonreentrant COBOL
RCOB - Reentrant ANS COBOL
PL1 - Nonreentrant PL/1
RPL1 - Reentrant PL/1
FORT - Fortran

LOADNAM

specifies the load module name, if the subsystem is to be dynamically loaded (that is, it is linkedited as a separate load module). If this parameter is specified, the SBSP parameter need not be specified (SCTNAME)(SCTSW1(SCTLOAD=1 if specified)).

LOG

specifies whether or not system log entries are to be made for this subsystem. Code NO to bypass the log facility. The default is YES. (SCTSW1(SCTLOG,YES=1,NO=0))

LSYNCH

specifies whether or not log records for this subsystem are critical, that is, must be added to the current buffer and written onto the system log immediately. Otherwise, the log records are accumulated until the buffer is full. YES specifies that records are to be written immediately; NO specifies that records are to be added to the current log buffer. The default is NO. (SCTSW1(SCTASYNC,YES=0,NO=1))

MNCL

subsystem-associated instruction:

For core-resident or dynamically loadable reentrant subsystems, the value delimits the number of possible concurrent subsystem executions; for nonreentrant, code 1.

For Overlay Region A reentrant subsystems, specifies the number of possible concurrent subsystem executions and the maximum number of executions that can be undertaken every time the overlay is loaded into main storage. For nonreentrant subsystems, specifies the maximum number of subsystem executions that can be serially undertaken every time the overlay is loaded. (SCTMNCL)

region-associated instruction:

The default and only valid value is 1.

NUMCL

indicates the number of entries that are to constitute the core queue of messages held in main storage. This parameter is not valid if BLRI=NQ is coded. The default is 0 (no core queue). (SCTBCLP,SCTECLP)

OVLY

subsystem-associated instruction:

indicates whether or not the subsystem is core resident or resides in overlay region A. If the subsystem resides in Overlay Region A, code the particular region A group it belongs to, that is, 4-62. The default is 0, indicating that the subsystem is core resident or dynamically loaded. There should be no gaps in the ascending sequence of OVLY numbers specified in the SYCTTBL macros for the Subsystem Control Table. For VS systems, see also the EXGRP parameter.

region-associated instruction:

indicates the overlay region with which the SYCTTBL instruction is to be associated. Code 1 for region B; 2 for region C; 3 for region D. (SCTPONU, bits 2-7.)

PCEN

specifies the percentage of the disk queue data set to be allocated to the terminal, subsystem or region. This data set is named via the DFLN parameter. PCEN is valid only if DFLN is coded. If a whole percentage value is to be specified, code units as a decimal value, range 1 to 100. If a fractional percentage value is to be specified, code (units,hundredths) where units and hundredths each are a decimal value within the range 0 to 99. For example, code PCEN=64 for 64 percent, PCEN=(0,80) for 0.8 percent. The default is 100 (100 percent). (SCTPCEN)

PL1

designates the compiler level under which the PL/1 program was compiled. The default is the standard 'F' level compiler. PL1=OPT must be specified if the program was compiled under the optional PL/1 Optimizing compiler. This parameter is valid only when the subsystem is written in PL/1. (SCTSW1(SCTPL1OP,OPT=1,F=0))

PL1LNK=

is coded PL1LNK=BASED if the program uses the non-standard 'BASED' linkage techniques in which parameters are received as dummy arithmetic scalars (refer to the PL/1 Programmers Guide). It is not needed if parameters are received as standard character strings. The default is NONBASED. This parameter is valid only for PL/1 programs, compiled under the PL/1 Optimizing compiler. (SCTSW1(SCTPL1LK,BASED=1,NONBASED=0))

PRYMSGS

indicates the number of entries for a core queue of priority messages (that is, messages with a "P" in MSGHUSR). Messages of this type are retrieved for processing before other messages in the normal core or disk queues. When the priority message queue is full, priority messages are queued on the normal message queue. The default is 0. (SCTPRNDX(relative pointer to priority queue chain))

PRTY

specifies the execution priority to be applied to this subsystem by the Dispatcher. Priority is coded as a value from 0 (highest) to 3 (lowest). The default is 0. (SCTPONU, bits 0,1)

RESOURC

specifies a resource identifier as defined by a RESOURCE macro ID parameter. This parameter limits the maximum number of concurrent threads that the Subsystem Controller starts for this resource. If this parameter is coded, more than one SYCTTBL may have the same RESOURC value. The RESOURCE macro must occur prior to all SYCTTBL macros in the same assembly. (SCTRESRC)

RESTART

specifies whether or not message restart is required. YES (default) indicates that it is. NO specifies that no message restart is to be performed. IFPOSBL indicates that message restart is not critical, but should be performed if the message is encountered within the system log read back point in time. (This specification does not affect messages encountered on the system log that had successfully completed processing prior to the time of system failure.) A subsystem which performs updates to a data base should specify YES. (SCTSW1(SCTRESTR,IFPOSBL or YES=1,NO=0; SCTRESTA,YES=1,IFPOSBL or NO=0))

REUSE

indicates whether the loaded subsystem is at least reusable (reusable or reentrant). If NO is coded, a fresh copy of the subsystem is loaded for each message; MNCL should be specified as 1 and BLDL=NO must be coded. This parameter is only valid if the LOADNAM parameter is specified. The default is YES, specifying that an existing copy of the loaded subsystem may be used for additional messages. See also the LANG and MNCL parameters. (SCTBIT3(SCTREUSE,YES=1,NO=0))

RVFILE

specifies whether or not a subsystem alters (updates, adds to, or deletes from) a file that is to be recovered (REVERSE=YES or REVERSE=CRITICAL coded as a FAR parameter) or alters areas of storage being checkpointed (such as USERSPA). The default is NO. (SCTBIT1(SCTDBUPD,YES=1,NO=0))

SBSP

subsystem-associated instruction:

specifies the label identifying the entry-point to be given control when work is to be forwarded to the subsystem. (Not required if LOADNAM is specified.)

region-associated instruction:

specifies the Intercomm entry-point MONOVLY. Entry points of subsystems residing in overlay regions B, C, and D are specified to the system via the OVLYBTB, OVLYCTB, and OVLYDTB tables described in the Operating Reference Manual. (SCTSBSP)

SCHED

specifies whether or not work for the subsystem or region is to be scheduled. Code NO if work is merely to remain queued (see the THRSH parameter). The default is YES, indicating that work is to be automatically forwarded. (SCTBIT1(SCTNOSCH,YES=0,NO=1))

SECU

specifies the index value of the user security routine (under Basic Security) to be given control before work is passed to the subsystem or region. Code as a decimal value, 1 to 63. The default is 0, which indicates that no security processing is required. The user security routines are accessed via the SECURITY table. The value ((index-1)*4) expresses the offset into the table for locating the security routine's V-type address constant. (SCTSECUR, bits 2-7)

SEGREST

specifies how message restart is to handle segmented messages sent by the subsystem to the Output Utility or Change/Display. YES causes the segment stream to be treated like a single message: if any of the segments are not completely processed, all the segments are restarted. NO (default) causes the segments to be treated like separate unrelated messages, and only the ones that are not completely processed are restarted. This parameter is only effective if RESTART=YES or RESTART=IFPOSBL. (SCTBIT2(SCTSGRST,YES=1,NO=0))

SNAP

specifies whether or not a snap (ID=118) is to be taken if a subsystem execution exceeds the time value assigned to the TCTV parameter. Code NO if a snap is not to be taken. The default is YES. (SCTBITS(SCTSNAP,YES=0,NO=1))

SOSO

specifies whether or not operator sign-on/sign-off under Basic Security is required by the subsystem or region. Code YES if sign-on/sign-off security is required. The default is NO. Refer to the SPALIST macro SONOFF parameter and the STATION macro UNIVER and OPER parameters. (SCTSECUR(SCTSECSG,YES=1,NO=0))

SPAC

subsystem-associated instruction:

specifies, in bytes, the amount of dynamic core (subpool 0) that is required per execution by the subsystem. When defining PL/1-optimized subsystems, specifies the amount of core to be dynamically acquired by the PL/1 interface (PREPLI) for use by the subsystem. (Refer to Operating Reference Manual.)

region-associated instruction:

specifies the maximum amount of dynamic core that is required per execution by a single subsystem within the region.

The amount of core is considered to be the amount that must be available before control can be passed to a subsystem (see also GET parameter). It is to exclude the length of any message forwarded, although if the processing subsystem employs the Edit Utility, this amount should be capable of absorbing the length of the edited message. Code as a decimal value, 0 to 524281. The default is 5000. If the Intercomm pools are correctly tuned so that the required storage is usually obtained from them, an amount as low as 2K (2048) may be specified; under MVS, 2K is recommended (except for PL/1-optimized subsystems). (SCTSPACE)

SRESP

specifies, for SNA teleprocessing systems, the type of response to be requested for output to a LU. NOSPEC means that this is to be determined from other sources. The default is (E,1). See SNA Terminal Support Guide. (SCTFLAGS(SCTFSDR, D=1, E or NOSPEC=0; SCTFSER, E=1, D or NOSPEC=0; SCTFSR1, response type of 1=1, 2 or NOSPEC=0; SCTFSR2, response type of 2=1, 1 or NOSPEC=0))

SUBC

specifies the low-order entry code to be used in the identification of the table entry. Code as either a single alphanumeric or special character, or as a three-digit decimal number 000 to 255. Note that the binary representation of a code of 1 is X'F1', while that of a code of 001 is X'01'.

The following high/low order table entry code combinations are reserved: GP and all codes whose high order code is 000 and whose low order code is within the alphabetic range A-Z. Refer to Operating Reference Manual for a list of code combinations used by Intercomm-supplied subsystems. (SCTSUBC)

SUBH

specifies the high-order entry code to be used in the identification of the table entry. Code as either a single alphanumeric or special character, or as a three-digit decimal number 000 to 255. Note that the binary representation of a code of 1 is X'F1', while that of a code of 001 is X'01'. (SCTSUBCH)

TCTV

subsystem-associated instruction:
specifies, in seconds, the length of time a single subsystem pass is permitted to execute.

region-associated instruction:
specifies, in seconds, the maximum execution time of any subsystem within the region.

This duration must include all I/O time, and potential enqueue wait time. Code as a decimal value, 1 to 27962. (If an execution exceeds this value, the Subsystem Controller assumes control, cancels the message, and logs the header with an X'FD' in the MSGHLOG field.) Refer to the CANC and SNAP parameters. The default is 60. 0 indicates that loop detection is not required; timing is not to be done. (SCTTCTV)

THRSH

specifies the number of messages that must be queued for the subsystem or region before any work is to be forwarded to the subsystem or region. Code as a decimal value, 0 to 255. The default is 0, indicating that work is to be forwarded upon arrival. (SCTHRESH)

TISE

specifies whether or not transaction security under Basic Security is required by the subsystem or region. Code YES if transaction security is required. The default is NO. Refer to the SPALIST macro TRANSEC parameter and the STATION macro VERBS parameter. (SCTSECUR(SCTSECVB,YES=1,NO=0))

TYPE

denotes whether a Subsystem Control Table entry is being generated, or a BTAM Front End Queue Control Table entry is being generated. Code INT for a Subsystem Control Table entry. Code BTAM for a BTAM Front End Queue Control Table entry. (Refer to the BTERM macro QNUM parameter.)

If TYPE=INT, all macro parameters are valid. If TYPE=BTAM, only the core and disk queue specification parameters are valid. The default is INT.

WTO

specifies whether or not a message, consisting solely of the uninterpreted bit configurations of the high and low order subsystem or region codes, is to be sent to the system console every time work is forwarded to the subsystem or region. Code NO if they are not to be sent. YES (default) is generally used for test mode. Some of these codes may not be printable. Printing of the message may also be suppressed by omitting the WTO parameter from the EXEC statement. (SCTBIT1(SCTNOWTO,YES=0,NO=1))

TMZONE -- Specify Clock-dependent Processing

The TMZONE macro instruction is used to create the PMITIMTB table entries that furnish INTERCOMM with the information required to associate any subsystem execution with any specific time of day. A single TMZONE macro instruction effectively prescribes a request for the execution of the designated subsystem at the designated time, passing that subsystem in both the header and text portion of the message generated macro information identifying the specific request.

The form of the TMZONE macro instruction is as follows:

(blank)	TMZONE	SCHT=time, PGID=lo-order-subsys-code, PGIH=hi-order-subsys-code, [TCTV= {subsys-execution-time / 30} ,] PVMI=user-code,TMZC=user-code

SCHT=
specifies the time of day at which the processing request is to be constructed. Using a 24-hour notation, code as 'hhmm' where "hh" represents the hour and "mm" the minute. (E.g., 4:25 p.m. would be encoded as 1625.)

PGID=
specifies the low-order subsystem code of the subsystem that is to receive the processing request. This code must be identical to that assigned to the associated subsystem's SYCTTBL macro instruction "SUBC=" parameter. Code accordingly as a single character or as a three-digit decimal number, range 000 to 255.

PGIH=
specifies the high-order subsystem code of the subsystem that is to receive the processing request. This code must be identical to that assigned to the associated subsystem's SYCTTBL macro instruction "SUBH=" parameter. Code accordingly as a single character or as a three-digit decimal number, range 000 to 255.

TCTV=

specifies, in seconds, the duration of time that is to override the associated subsystem's subsystem control table SYCTTBL macro instruction "TCTV=" value. This duration will override the normally delimiting time unit for the time-dependent processing request only.. Code as a decimal value, 1 to 27962. The default code is 30.

PVMI=

specifies a code that may be used to identify the request. This code may qualify the code assigned to the "TMZC=" parameter (or vice versa); however, the particular use of this field is user-designated. Code as a decimal value, 0 to 255.

NOTE: This code will be inserted in the message header field, MSGHVMI, when the request message is constructed.

TMZC=

specifies a code that may be used to identify the request. This code may qualify the code assigned to the "PVMI=" parameter (or vice versa); however, the particular use of this field is user-designated. Code as a decimal value, 0 to 255.

NOTE: This code will be inserted in the third byte of the text portion of the message when the request message is constructed.

USRTRACK -- Track User Data Using SAM

The USRTRACK macro is used in conjunction with the Systems Accounting and Measurement (SAM) facility to track user data. Separate USRTRACK macros are coded to either increment a user-defined MAPACCT accumulator by one (see MAPACCT macro) or invoke a user written SAM exit routine. (See the Operating Reference Manual.)

The form of the USRTRACK macro instruction is as follows:

symbol	USRTRACK	{BUCKTNO=n} {FUNCNO=n } [,SPA=(r)]

BUCKTNO
: is the accumulator (bucket) number to be incremented by one. Code as a two-digit decimal number from 01 to 10, inclusive.

FUNCNO
: is the SAM user exit routine number to be invoked. Code as a two-digit decimal number from 01 to 10, inclusive.

SPA
: specifies a register containing the address of the SPA. If this is specified, the address of the SAMTRACK function will be obtained from the SPA. Otherwise, a V-type address constant will be used.

Note: BUCKTNO and FUNCNO are mutually exclusive operands; code one or the other, but not both.