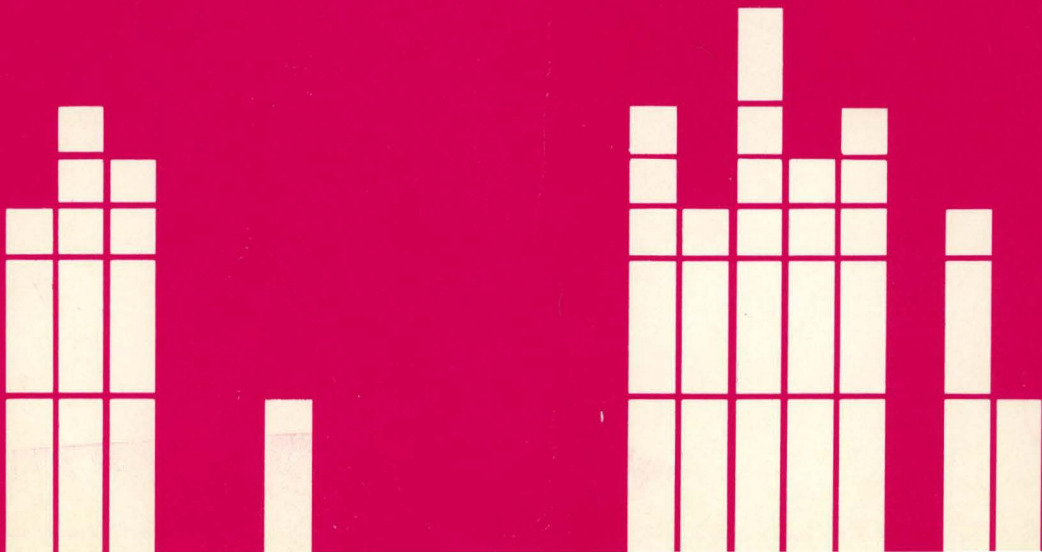


4700 Finance
Communication System

Controller
Programming Library

Volume 6
Control Program
Generation

IBM



4700 Finance
Communication System

Controller
Programming Library

Volume 6
Control Program
Generation

Publication Number
GC31-2071-1

File Number
S370/4300/8100/S34-30

Second Edition (January 1984)

This edition applies to the Release 3 version of the IBM 4700 Finance Communication System current with the date above, and to all subsequent versions and releases until otherwise indicated in new editions or Technical Newsletters.

This publication replaces *IBM 4700 Finance Communication System Local Configuration Facility User's Guide*, GC31-2031-0 entirely. It also replaces and obsoletes certain information (particularly Chapter 11) of *IBM 4700 Finance Communication System Principles of Operation*, GC31-2029-1.

Changes to the information herein can occur often; before using this publication in connection with the operation of IBM systems, consult the latest *IBM System/370 Bibliography of Industry Systems and Application Programs*, GC20-0370, for the editions that are applicable and current.

A revision bar (solid vertical line) appears in the left margin adjacent to any new or changed information produced as a result of revising the publication for this edition.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this publication is not intended to state or imply that only IBM's program product can be used. Any functionally equivalent program can be used instead.

Publications are not stocked at the address below; you should make any requests for IBM publications to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is at the back of this publication. You can use this form to send IBM any comments or suggestions, or, if the form is already gone, you can mail your comments directly on another sheet of paper to IBM Corporation, Information Development, Department 78C, 1001 W. T. Harris Boulevard, Charlotte, NC 28257, U.S.A. IBM may use or distribute the information you supply in any way it believes appropriate without incurring any obligation to you.

Preface

This is Volume 6 of the *4700 Controller Programming Library* — one of a set of six volumes for the 4700 programmer. Figure 0-1 on page v summarizes the topics covered in this and the other volumes. All six volumes are available from your IBM representative or local branch office under a single order number (GBOF-1387).

You need this volume if you are involved with generating an operational 4700 system (the *control program*, or *controller configuration*) for a 4700 controller. This process — and its resulting output — are known colloquially as a *CPGEN*.

What This Book Is About

This publication explains the concept and process of *CPGEN* and gives reference information for the commands and macro instructions for doing one. It also describes the local configuration facility (LCF), which you can use to generate a controller configuration at the controller itself.

Who Should Read this Book

People who need to know how to do a *CPGEN* for a 4700 controller should read this book. If you are such a person, your job title is probably something like “system programmer” or “systems analyst”.

How this Book Is Organized

This book has four chapters and three appendixes:

- Chapter 1 describes the overall process for doing a *CPGEN*.
- Chapter 2 gives a sample *CPGEN* and explains its makeup.
- Chapter 3 gives the reference information for the instructions and macros you use to perform *CPGEN*.
- Chapter 4 describes how to use the local configuration facility.
- Appendix A contains the *CPGEN* messages in numeric sequence, along with explanations and recommended actions.
- Appendix B contains the LCF messages in numeric sequence, along with explanations and recommended actions.
- Appendix C contains the macros and macro operands uniquely used for generating the Alternate Line Adapter Feature (RPQ 8V0132).

What Else to Read

If your 4700 system has not yet been started, read the *IBM 4700 Finance Communication System Subsystem Operating Procedures*, GC31-2032 (a prerequisite), for a description of how to log on and start the 4700 system. Then return to this book for guidance in using CPGEN and LCF.

For any other information regarding programming for the 4700 controller, refer to various parts of the *IBM 4700 Controller Programming Library*:

Volume 1: General Controller Programming, GC31-2066.

Volume 2: Disk and Diskette Programming, GC31-2067

Volume 3: Communication Programming, GC31-2068.

Volume 4: Loop and DCA Device Programming, GC31-2069.

Volume 5: Cryptographic Programming, GC31-2070.

For information describing diskette formats, see:

IBM Diskette General Information Manual, GA21-9182.

For information describing how to design and code a host application program, using VTAM, to transfer files from the host, see:

IBM ACF/VTAM Macro Reference, SC38-0261

IBM ACF/VTAM Macro Language Guide, SC38-0256

For information regarding storage requirements and how to determine them, refer to:

IBM 4700 Finance Communication System: System Configurator, GC31-2017.

<p>VOLUME 1: GENERAL CONTROLLER PROGRAMMING (GC31-2066)</p> <ul style="list-style-type: none"> • Coding Considerations • Instruction Categories • Coding Rules • Instruction Descriptions • Machine Instruction Formats • Copy Files • Error Messages, Program Check Codes, and Status Codes • Functions Retained for 3600 Compatibility • Program Communication with the System Monitor
<p>VOLUME 2: DISK AND DISKETTE PROGRAMMING (GC31-2067)</p> <ul style="list-style-type: none"> • Basic Disk and Diskette Programming • Extended Disk and Diskette Access Method • Instruction Descriptions • Machine Instruction Formats • Copy Files • Status Codes
<p>VOLUME 3: COMMUNICATION PROGRAMMING (GC31-2068)</p> <ul style="list-style-type: none"> • SNA/SDLC Communication Programming • SNA/SDLC Communication Macros • SNA/SDLC Communication Instructions (Reference) • BSC3 Host Communication Programming • BSC3 Communication Instructions (Reference) • Communication Status Codes • Communication Parameter List Reference • Communication Machine Instruction Formats
<p>VOLUME 4: LOOP AND DCA DEVICE PROGRAMMING (GC31-2069)</p> <ul style="list-style-type: none"> • General Protocols for Displays • 4704 and 3604 Displays • 3270-Compatible Displays and Devices • 3606 and 3608 Financial Services Terminals • General Protocols for Printers • 4710 and 4720 Printers • 3610, 3611, and 3612 Printers • 3615 and 3616 Printers • 3270-Compatible Printers • 3624 Consumer Transaction Facilities • Data Stream Mapping (DATSM) Protocols • Device Status Codes • Device Parameter List Reference
<p>VOLUME 5: CRYPTOGRAPHIC PROGRAMMING (GC31-2070)</p> <ul style="list-style-type: none"> • Cryptographic Concepts and Facilities • Enciphering and Deciphering Operations • Generating and Exchanging Cryptographic Keys • Authenticating Messages • Validating and Translating PINs • Using the Encrypting PIN Keypad • Host Support Encryption Routines (BDKDPRS and BDKDES) • Cryptographic Programming Instructions (Reference) • System Cryptography • Cryptographic Machine Instruction Formats • Cryptographic Parameter List Reference • Cryptographic Program Checks and Status Codes
<p>VOLUME 6: CONTROL PROGRAM GENERATION (GC31-2071)</p> <ul style="list-style-type: none"> • Overall View of Control Program Generation (CPGEN) • Sample CPGEN • CPGEN Macro Statements (Reference) • Using the Local Configuration Facility (LCF) • CPGEN Messages • LCF Messages

Figure 0-1. 4700 Controller Programming Library (GBOF-1387)

Summary of Amendments

This edition replaces GC31-2071-0.

This edition — for Release 3 of the IBM 4700 Finance Communication System — incorporates information to reflect the following enhancements to the system and CPGEN:

- Additional disk drives and multiple-sector formatting
- Expanded controller storage capability
- Alternate-Line Adapter (ALA) macros
- User-defined data sets.

The 4700 system supports as many as two disk drives, designated by the letters A and B.

Contents

Chapter 1. Introducing the Control Program Generation Process	1-1
Using CPGEN at the Host, in Brief	1-2
Using LCF at a Controller, in Brief	1-2
Coding the Configuration Macro Instructions	1-3
Coding STARTGEN and ENDGEN	1-4
Coding LOOPS	1-8
Coding OUTRTBL	1-8
Coding TRTBHDR	1-8
Coding INTRTBL	1-8
Coding MSTRTBL	1-8
Coding DCAxxxx	1-8
Coding DEVxxxx	1-8
Coding DCAPORT	1-9
Coding DEFADDR	1-9
Coding LDA	1-9
Coding COMLINK	1-9
Coding FILES	1-9
Coding SEGSTOR	1-9
Coding STATION	1-9
Coding OPTMOD	1-9
Coding APLIST	1-10
Coding EXTCTR	1-10
Coding DEFSPA	1-10
Coding PINTBL	1-10
Coding OUTSPEC	1-10
Coding PSSLST	1-10
Coding TRANPL	1-11
Coding DCATRTBL	1-11
Coding TRTLIST	1-11
Coding DEFSBF	1-11
Coding DKATBL	1-11
Considerations for Changing Over from a 3600 System	1-12
Storage Allocation	1-12
Unsupported Configuration Macros	1-12
Selecting Optional Modules	1-12
Defining 3278 Shift Keys	1-13
System Monitor Requirements for Using the Host Transmission Facility	1-13
Chapter 2. Examining a Sample CPGEN	2-1
Introducing Configuration Concepts	2-1
Examining the Configuration Code	2-5
Chapter 3. Using the Configuration Macros	3-1
Interpreting the Syntax Notation	3-2
APLIST – Add Application Programs to Configuration List	3-3
APMEDIA – Specify Application Program Residence	3-4
CHARDEF – Define User-Specified Special Characters	3-5
COMLINK – Define Communication Link	3-7
DCAPORT – Specify a Device Cluster Adapter Port Assignment	3-14
DCATRTBL – Define a DCA Translation Table	3-15
DCA3262 – Define a 3262 Printer	3-17
DCA3278 – Define a 3278 Keyboard Display	3-18
DCA3279 – Define a 3279 Keyboard Display	3-20
DCA3287 – Define a 3287 Printer	3-23
DCA5210 – Define a 5210 Printer	3-24
DCA4704 – Describe a 4704 Display	3-25
DEFADDR – Specify the Device Address	3-28
DEFSBF – Define System Image Buffers	3-30
DEFSPA – Define Scratch Pad Area	3-31
DEV3604 – Describe a 3604	3-32
DEV3606 – Describe a 3606	3-35
DEV3608 – Describe a 3608	3-36
DEV3610 – Describe a 3610	3-38
DEV3611 – Describe a 3611	3-40
DEV3612 – Describe a 3612	3-42
DEV3614 – Define a 3614	3-45

- DEV3615 -- Define a 3615 3-46
- DEV3616 -- Define a 3616 3-48
- DEV3624 -- Define a 3624 3-52
- DEV4704 -- Describe a 4704 3-53
- DEV4710 -- Describe a 4710 3-56
- DEV4720 -- Describe a 4720 3-59
- DKATBL -- Specify a Dead-Key Table for Accented Characters 3-62
- Examples 3-63
- ENDGEN -- End Configuration Program Generation 3-64
- EXTCTR -- Specify Extended Statistical Counters 3-66
- FILES -- Specify Diskette or Disk Files 3-68
- INTRTBL -- Build Input Translation Table 3-73
- LDA -- Specify the Logical Device Address 3-77
- LOOPS -- Define Terminal Loops 3-80
- MSTRTBL -- Build Magnetic Stripe Translation Table 3-81
- Programming Notes for 3278 and 3279 Magnetic Stripe Readers 3-83
- OPTLIST -- Specify Optional Module List 3-84
- OPTMOD -- Specify Optional Module List 3-85
- OUTBHDR -- Define Loadable World Trade Special Character Tables 3-87
- OUTRTBL -- Build Output Translation Table 3-88
- OUTSPEC -- Define a World Trade National Use Differences Table Character Set 3-90
- PINTBL -- Describe a PIN Keypad Translation Table 3-93
- PRIDSP -- Priority Table Definition 3-95
- PSSLST -- Define Programmed Symbol Sets 3-97
- SEGINIT -- Initialize Storage Segments 3-98
- SEGSTOR -- Specify Segment Storage 3-100
- STARTGEN -- Start Controller Configuration 3-101
- STATION -- Specify a Station 3-106
- TRANPL -- Define a Storage Pool 3-114
- TRTBHDR -- Specify Translation Table Headers 3-116
- TRTLIST -- Specify a Translation Table List 3-118

Chapter 4. Using the Local Configuration Facility -- LCF 4-1

Introducing LCF 4-1

- CPGEN, What is it 4-1

- Using LCF Procedures 4-1

- Creating the CPGEN File 4-2

- Editing the CPGEN Information 4-3

- Transferring Application Programs 4-3

- Transferring an Entire CPGEN File 4-3

- Maintaining a CPGEN Library 4-3

- Using LCF 4-3

- Preparing an LCF Generation 4-4

- Checking for the LCF Prerequisites 4-4

- 50-Key Keyboard Support 4-4

- Optional Printer 4-5

- Library Diskette Support 4-5

- Using LCF 4-5

- Starting LCF 4-6

- Creating the CPGEN File 4-7

- Entering Comments 4-9

- Correcting Errors 4-10

- Using Continuation Cards 4-11

- Ending CPGEN Entry 4-11

- Storing Files on the Library Diskette 4-11

- Editing the CPGEN File 4-12

- Correcting a CPGEN Statement 4-15

- Adding a CPGEN Statement 4-17

- Deleting a CPGEN Statement 4-17

- Searching for Statements by Type 4-18

- Duplicating a CPGEN Statement 4-19

- Searching for a Statement Label 4-20

- Repeating Search Commands 4-21

- Printing the CPGEN Statement File 4-21

- Leaving the LCF Editor 4-22

- Creating an Operating Diskette 4-22

- Transferring Programs via Diskette 4-25

- Allocating Space for Application Programs 4-26

- Transferring All Application Programs 4-28

- Replacing Application Programs 4-29

- Adding Application Programs 4-31
- Receiving a CPGEN File 4-33
 - Receiving CPGEN Files on a Basic Exchange Diskette 4-34
 - Receiving CPGEN Files through the Host Link 4-34
- Using LCF Panels 4-35
 - Using the 4700 LCF Program Menu 4-35
 - Changing the LCF Password 4-35
 - Invoking the System Monitor 4-36
 - Using the Main LCF Menu 4-36
 - Defining a Configuration Panel 4-37
 - Statement Entry Panel 4-37
 - Continuation Panel 4-37
 - Edit Panel 4-38
 - Transfer AP Panel 4-39
 - Receive CPGEN File Panel 4-40
- LCF Interfaces and Limitations 4-41
 - LCF Interfaces 4-41
 - CPGEN File from Diskette 4-41
 - Transferring CPGEN Files from the Host 4-41
 - Transmission Protocols 4-43
 - Transferring Programs by Diskette 4-44
 - Maximum CPGEN File Sizes 4-45
 - LCF Limitations 4-46
- Using the Library Diskette Statement File 4-48
 - Creating Library Diskette Statement Files 4-48
 - Creating Configuration Data Set Files 4-49
- Migrating from Release 1.0 to Subsequent Releases 4-49

Appendix A. CPGEN Messages A-1

- General CPGEN Messages A-1
- Alternate Line Adapter Messages A-20

Appendix B. LCF Messages B-1

Appendix C. Alternate Line Attachment Feature Configuration Macro Descriptions C-1

- ALABUFF – Create Buffers C-2
- ALALINE – Define ALA Line C-3
- ALACU – Define ALA Control Unit C-6
- ALATERM – Define ALA SLU C-8
- ALA Changes to 4700 CPGEN Macro Instructions C-10
 - STARTGEN Macro C-10
 - STATION Macro C-11

Index X-1

Figures

- 0-1. 4700 Controller Programming Library (GBOF-1387) v
- 1-1. Configuration Macros Coded BEFORE Writing Application Programs 1-5
- 1-2. Configuration Macros Coded AFTER Writing Application Program 1-6
- 1-3. Coding the STATION Macro 1-7
- 2-1. Overall Physical Configuration 2-2
- 2-2. Logical Layout (Configuration) of Work Stations 2-4
- 3-1. Permissible CHARSET Parameters for the OUTRTBL Macro 3-89
- 4-1. 50-Key Numeric/Function Keyboard as Interpreted by LCF 4-5

Chapter 1. Introducing the Control Program Generation Process

CPGEN (control program generation) is a process that uses a set of configuration macro instructions to identify and define the overall characteristics of your particular IBM 4700 Finance Communication System. These characteristics include:

- Those of each device
- Those of each work station (a collection of devices to be used as a group together)
- The physical and logical relationships among these pieces of hardware.

You must complete the CPGEN before your applications can run. That is, your 4700 operating environment does not exist until defined through the CPGEN process. The process defines the resources and relates them to the stations — that is, the logical grouping of application programs and system resources such as storage and devices — that will use them. It includes such things as allocating global and work-station storage, assigning logical addresses to terminals and devices, and defining characteristics of the host communication link.

The two basic ways of performing this process are to:

- Assemble an appropriate collection of CPGEN macros at the host operating system

-OR-

- Use the local configuration facility (LCF) at the controller for the same basic purpose.

This process creates a controller load image that is transmitted to the controller from the host, or that is written onto diskette using LCF. A host-created controller load image can reside either entirely on diskette, or partly on diskette and partly on disk. The parts that can be either on diskette or disk consist of the system monitor and application program data sets. The remainder of the controller load image must always be on diskette. An LCF-generated controller load image, however, must reside entirely on diskette.

Multiple application program data sets can coexist, and they can be on either disk or diskette, either individually or in multiple. Only one such data set can be active for a given IPL of the controller, however. You specify which data set this is to be, and where it resides, by using the APMEDIA macro in the CPGEN. APMEDIA can also denote where the system monitor data set is to be.

If you want to change which application program data set is to be active, you can do so either by:

- Using the system monitor's 078 command as described in *Subsystem Operating Procedures*.
- Using the deferred usage function of the host transmission facility's UPDATE command of the Host Support licensed product, as described in the *Host Support User's Guide*.

After doing so, you restart (IPL) the controller. Then the load image transfers from diskette or disk (or both) to controller storage, and the controller thus becomes functional with the newly-designated data set as the active one.

Using CPGEN at the Host, in Brief

This chapter describes the CPGEN process primarily as performed at the host, although the broad concepts apply equally well to using LCF. We outline here the use of some of the principal configuration macros without attempting to cover any particularly detailed considerations. LCF accepts these same macros prepared according to the same rules of syntax.

Then, in Chapter 2, we present a workable and comprehensive example of a CPGEN for a fairly small system (two display-keyboard devices and two printers arranged as two work stations). This example begins with an overall description and has explanations of specific points interspersed among the example's code.

Chapter 3 has all the detail you need for coding individual configuration macros. The descriptions there are primarily for reference. Each macro description does, however, begin with a statement of what the macro does or is to be used for. Included are special considerations for its use. These include such information as how various macros interact with one another. You'll find a layout of syntax rules and all the detailed parameter descriptions. Where applicable, the parameter descriptions include special notes and usage restrictions.

Using LCF at a Controller, in Brief

Chapter 4 explains how to use LCF to do a CPGEN at a controller. Two principal reasons for doing so are:

- Your 4700 system might not be connected to a host operating system.
- Even if connected, it might be worthwhile to generate at least some controller configurations locally, for whatever reasons.

If you use multiple 4700 controllers connected to a host and have the 4700 Host Support licensed program installed, you have yet another option. You can use LCF to generate a controller configuration at one of the controllers, then use Host Support's host transmission facility to receive the configuration image and retransmit it to other connected controllers. Doing this might help reduce necessary host computer time, which might be a premium resource.

Note, however, that LCF does not support the following 3600 or 4700 configuration macros or functions:

- EXTCTR
- SNA-Primary communication link operation
- PRIDSP
- Financial Services Terminals (3606 and 3608)

Coding the Configuration Macro Instructions

A 4700 operating environment results from defining what is known as a *controller load image* for use on an *operating diskette*. You insert the operating diskette into one of a 4700 controller's diskette drives to *program* the controller in the way it is to operate. Although oversimplified here, this process is basically how you get your 4700 system to be able to do useful work.

The definition for a controller load image consists of configuration macros and controller application programs. To define a controller load image at the host processor, you assemble these macros and programs and combine the resulting object modules. Then you can transmit this load image to the controller, thereby creating an operating diskette at the controller. When you start the controller, the load image transfers from the diskette to controller storage, which makes the controller functional.

You can also use the 4700's local configuration facility (LCF) to create the controller load image instead. LCF operates on the controller and accepts the configuration macro instructions in source form as input. LCF then generates the definition of the controller load image and writes it onto an IBM-supplied diskette, making it the operating diskette.

With LCF, the source macros can come directly from the operator using a 4704 display, from a diskette written in the basic exchange format, from an LCF library diskette, or they can be transmitted from the host. The source macro statements for use by LCF must conform to the same syntax as used by the host process.

The 4700 subsystem configuration macro instructions define the physical and logical characteristics of the subsystem. These macro instructions assemble into a configuration object module. A procedure for coding these macro instructions appears in Figures 1-1 through 1-3.

Figure 1-1 shows some configuration macros that you might code to define your configuration.

Figure 1-2 shows the macros that are easier to code **after** the application programs are written. They specify the diskette file allocations, controller storage allocations and initial values, and logical work station options.

Figure 1-3 shows how you code STATION in two steps when you use this procedure.

Coding STARTGEN and ENDGEN

STARTGEN is the first macro in the configuration deck. It specifies system characteristics and must be the first macro in any CPGEN.

ENDGEN is normally the last macro in the configuration deck. It signals the end of the configuration and provides a way to initialize station storage. That storage is arranged into segments.

If you need to initialize more segments than the ENDGEN macro can handle, you can code several SEGINIT macros **following** ENDGEN to do so. (This is the case where ENDGEN is not the last macro in a CPGEN.) You can use the SEGINIT macro in conjunction with the ENDGEN operand, or replace the *initialization* operand of ENDGEN. Code the SEGINIT operand just as you would code the ENDGEN operand.

Between STARTGEN and ENDGEN you code all of the other macros that your configuration requires. Some of the CPGEN macros — for example, STATION and FILES — are necessary in **any** CPGEN. Others are purely optional. For example, if your configuration is to use no DCA-attached devices, you obviously need not code any DCAXxxx macros to define such devices, nor need you code a DCAPORT macro to define attachment for them.

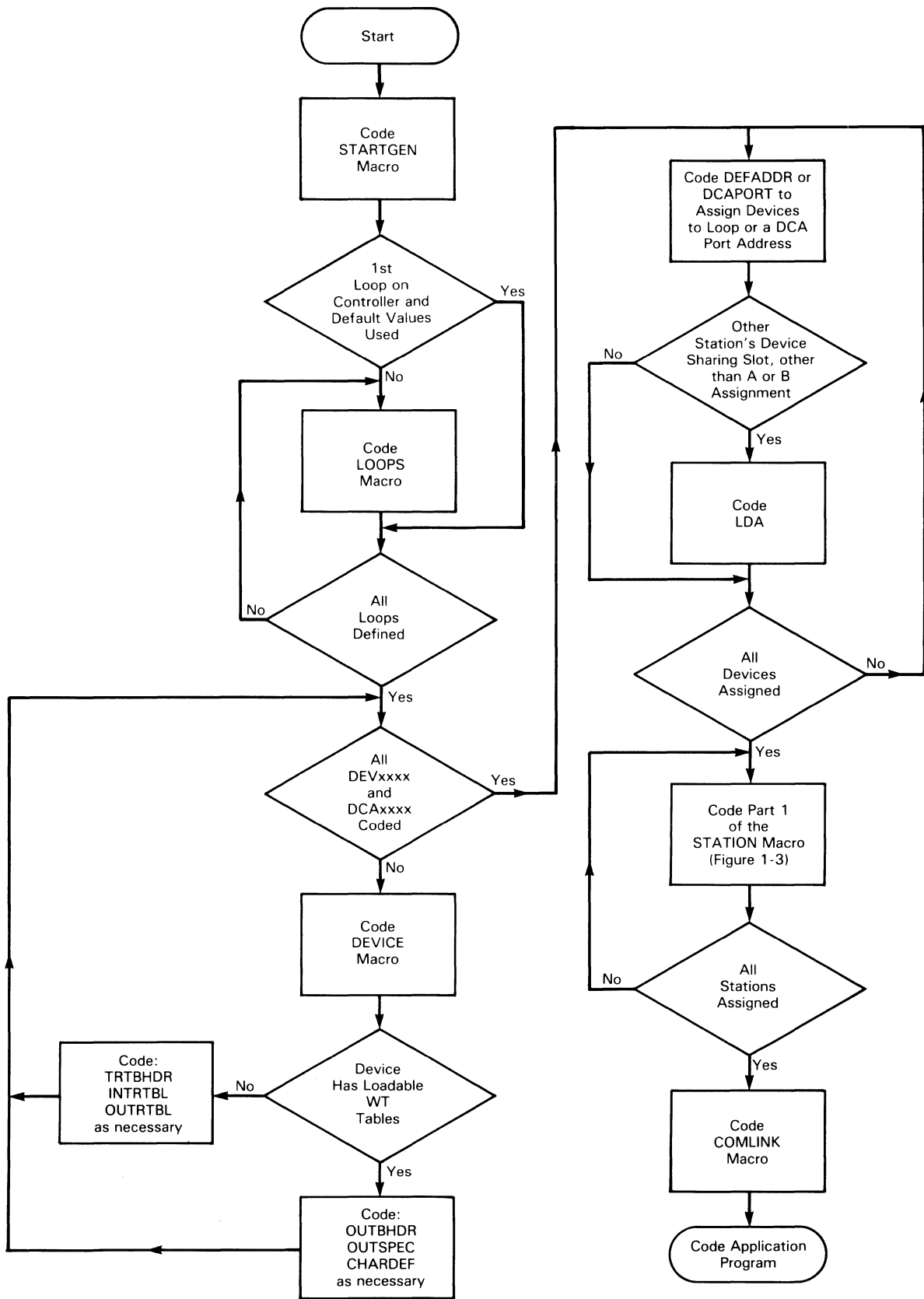


Figure 1-1. Configuration Macros Coded BEFORE Writing Application Programs

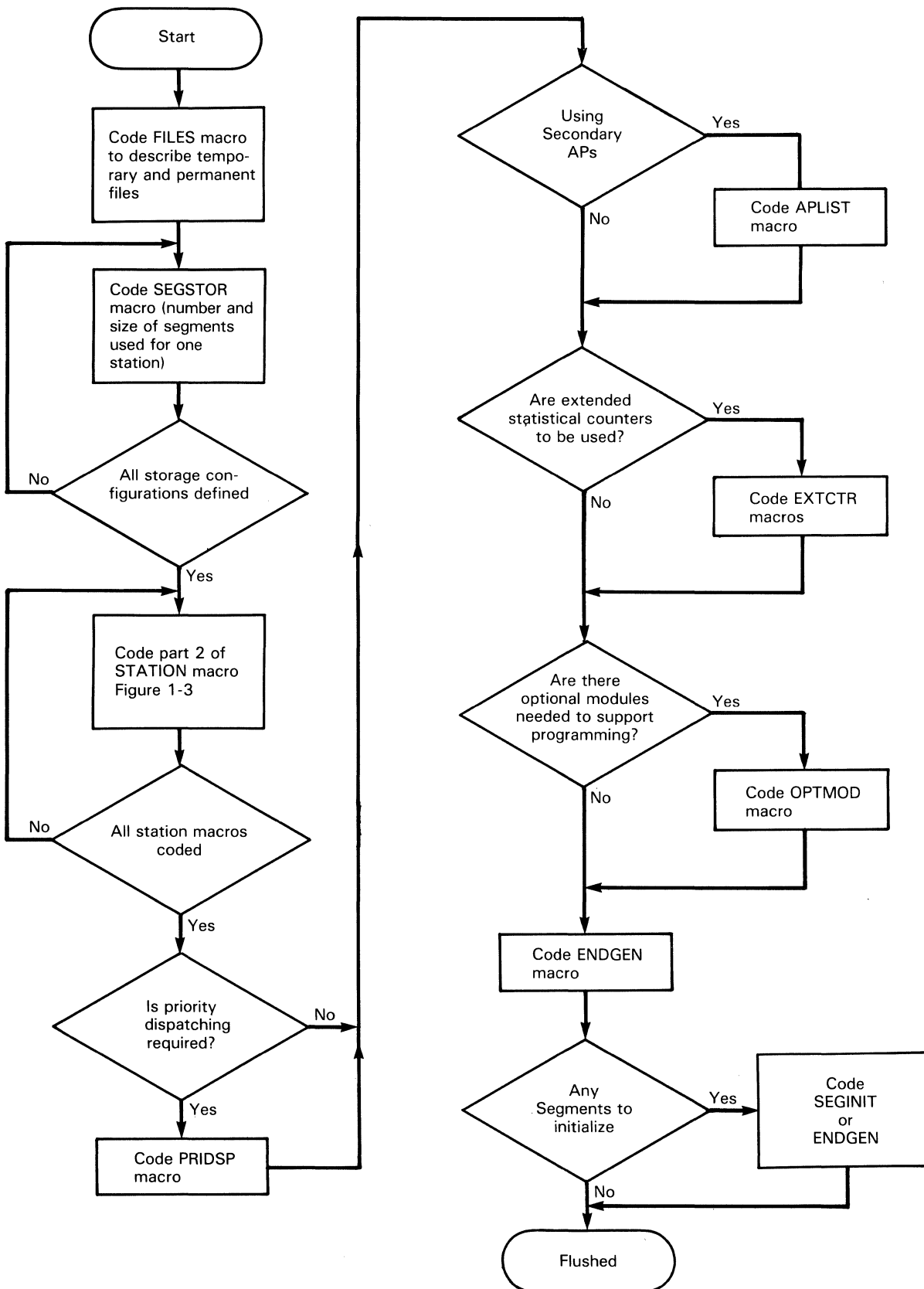


Figure 1-2. Configuration Macros Coded AFTER Writing Application Program

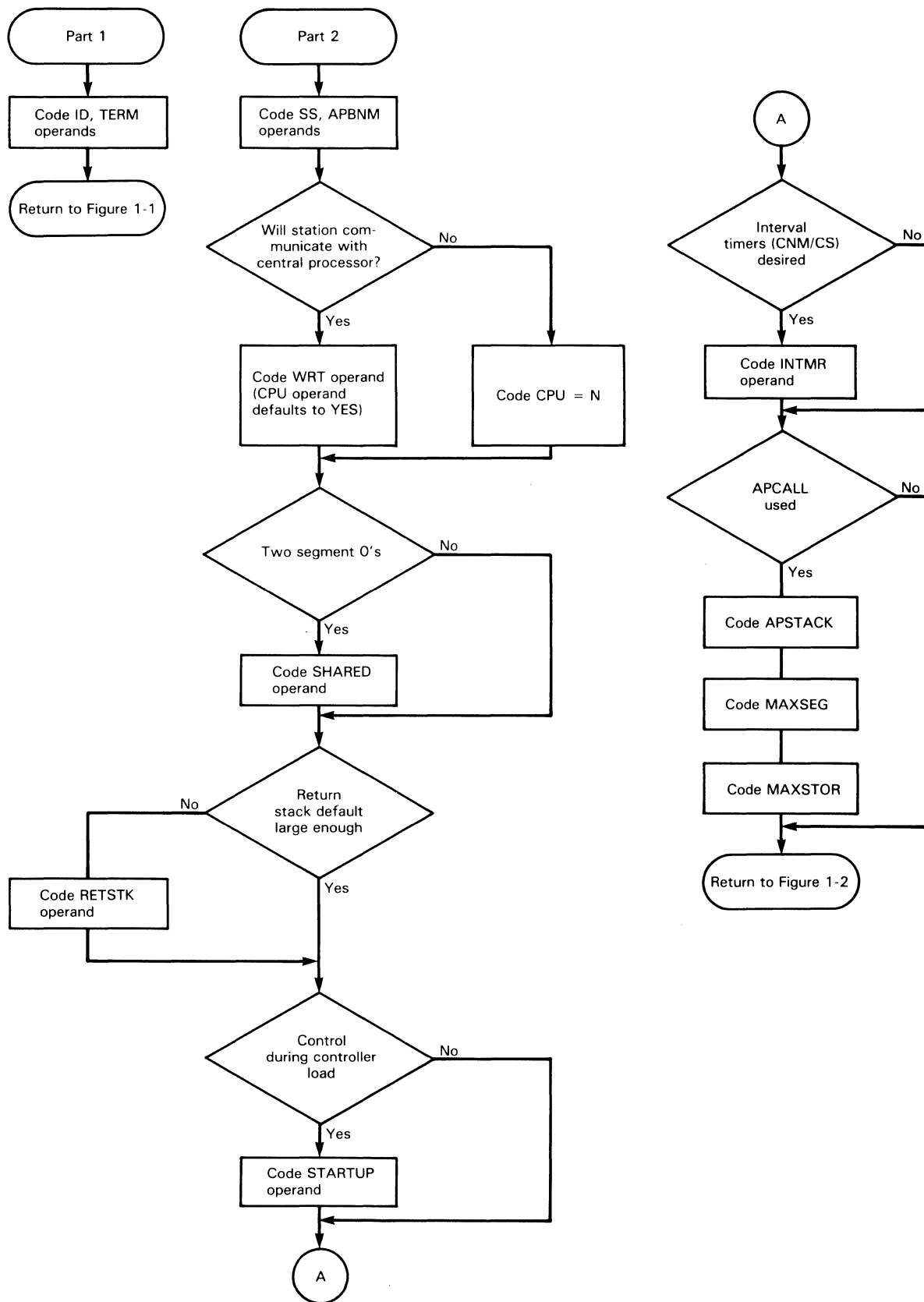


Figure 1-3. Coding the STATION Macro

Coding LOOPS

You code one LOOPS macro for each loop on the controller. LOOPS specifies the loop number and speed, and whether the loop is local or remote.

Coding OUTRTBL

You code one or more OUTRTBL macros depending on the number of output terminal components. Several components can use the same output translation table. Each OUTRTBL macro builds a 192-byte output translation table in the controller's table storage area so that all positions are invalid. The macro then modifies the table to include a standard character set. Finally the macro further modifies it to include the additional changes you set up in your OUTRTBL specification.

OUTRTBL specifies the terminal component type, standard translation table for the character set used by the component, and any modifications that are to be made to the translation table.

Coding TRTBHDR

TRTBHDR associates as many as four input translation tables with a single keyboard. This macro also defines a reset (cancel) key, nonlocking shift keys, and typematic (repeat-action) keys. The keys specified in TRTBHDR are common for all of the referenced input translation tables.

The user (a teller, quite likely) selects the use of a specific translation table through use of shift keys.

Coding INTRTBL

You can specify as many as four input translation tables for each terminal keyboard. The number depends on the data-entry needs of the teller. INTRTBL specifies a standard input translation table, the case it represents, and any modifications to the standard input translation table.

Coding MSTRTBL

MSTRTBL associates a special-purpose, or nonstandard, translation table for the magnetic stripe reader or magnetic stripe encoder. If the standard translation table is to be used with the magnetic stripe reader-encoder, you need not specify the MSTRTBL macro.

Coding DCAxxxx

A DCAxxxx macro, combined with the DCAPORT macro, describes the characteristics of a device attached to the device cluster adapter (DCA). The DCAPORT macro is selected by the DA operand of the STATION macro.

Coding DEVxxxx

The DEVxxxx macros describe the characteristics of terminals attached to controller loops. The LDA and DEFADDR macros refer to one or more DEVxxxx macros when they assign logical device and loop-slot addresses to devices. A DEVxxxx macro can be used by more than one loop-slot definition.

Coding DCAPORT

The DCAPORT macro logically assigns the terminal address to a port and identifies the unit address to the subsystem.

Coding DEFADDR

The DEFADDR macro logically assigns the terminal address to a slot in the transmission frame and identifies the unit addresses to the subsystem.

Coding LDA

LDA enables you to assign a device operating on one loop address to two different work stations. A device such as a 3616 passbook and document printer can be shared between work stations if you code a reference to different LDA macros in each STATION macro. Each LDA macro selects the same DEFADDR macro, which assigns the address, and the same DEVxxxx macro as that coded in the DEFADDR macro, which defines the device. The LDA macros assign logical device addresses for their appropriate work stations.

Coding COMLINK

COMLINK describes the communication link and certain features of the modem or communication adapter, such as the speed of the link, whether the modem has wrap capability, the number and size of the read buffers, and the number of write buffers.

Coding FILES

FILES identifies such things as the number and type of files on the diskette or disk, the indexing scheme used, and the amount of space available. One FILES macro is necessary for each drive. The space available on the primary diskette is occupied by the controller application programs, the permanent file, the temporary files, and user data sets.

Coding SEGSTOR

SEGSTOR specifies the number and size of the segments in station storage. Each specification is associated with a class, which you specify in the STATION macro.

Coding STATION

STATION creates a logical work station by associating terminals, storage, and an application program with each other. You code one STATION macro for each work station that is to be in your configuration.

Coding OPTMOD

OPTMOD enables the application programmer to include optional modules containing special programming facilities into the subsystem during the configuration procedure. If you don't use any of these functions, you need not code OPTMOD.

Make sure that an optional module is present, however, before running an application program that requires it. Otherwise, program checks can occur. Refer to "Selecting Optional Modules" later in this chapter for additional information about optional modules.

Coding APLIST

This macro defines the application programs called by the APCALL or DTACCESS facilities, but not defined by the STATION macros. *Controller Programming Library, Volume 1* has additional information about APCALL and DTACCESS.

Coding EXTCTR

EXTCTR defines a set of extended statistical counters for a device, a group of devices, or a loop. The extended counters record a count of total bytes transmitted and bytes in error for input devices, and noise errors for loops. The primary purpose of the extended counters is to provide statistics that can help in isolating degraded loop segments.

You assign the extended counter an external two-byte, hexadecimal identifier by using EXTCTR's ID parameter. This value determines the order in which the controller maintains the counters, and enables a STATS instruction (an execution-time instruction) to specify which counter is to be obtained or reset.

The EXTCTR instruction specifies which device, devices, or loops are assigned to which extended counters. Because the information in the counters is used in evaluating loop performance, it can suggest an appropriate number of extended counters for the system. For example, an extended counter might be defined for each modem location on a remote loop. Because one or more point-to-point circuits constitute a remote loop, defining an extended counter for each remote location helps to isolate degraded circuits on that loop.

Coding DEFSPA

The DEFSPA macro defines and identifies a scratch-pad area for expanded user storage. (To use the scratch-pad area, you must include optional module P2A on the OPTMOD macro.)

Coding PINTBL

PINTBL specifies the required translation table for an attached PIN keypad device.

Coding OUTSPEC

The OUTSPEC macro specifies one World Trade Table or National Use Graphics Table (as described in the *IBM 3270 Information Display System Character Set Reference*). Each of these character sets describes the special characters unique to each language. You must code one OUTSPEC macro for each language table you want. An OUTSPEC macro can be referred to by one or more OUTBHDR macros.

Coding PSSLST

PSSLST defines programmed symbol sets for devices that support them, and allows two such sets to be grouped together for a device. Programmed symbol sets are valid only for 3278 and 3279 Display Devices.

Coding TRANPL

TRANPL defines a storage pool, which can be one of two types:

- A *station* pool
- A *segment-class* pool.

To define a station pool (a storage pool for the use of a specific work station), you code the label of the TRANPL macro that defines that pool on the TRANPL operand of the STATION macro for the station that is to have the use of the pool. Multiple stations can share the use of a single pool if the STATION macro for each specifies the same label on its TRANPL operand.

To define a segment-class pool, you code the ID operand of the TRANPL macro and do **not** refer to the label of this TRANPL macro from any STATION macros.

Coding DCATRTBL

The DCATRTBL macro defines a translation table for a device attached through the device cluster adapter. This macro defines the standard translation table for as many as four shift cases plus the ALT case and any other modifications to the standard translation table. You use this macro only if you intend to use the 4704-2 device in its local-keytracking mode.

Coding TRTLIST

The TRTLIST macro associates as many as five internal translation tables with a single keyboard on a 4704-2 display-keyboard device in local-keytracking mode. You define these translation tables by using the DCATRTBL macro.

Coding DEFSBF

The DEFSBF macro instruction allocates a system image buffer to be used to transmit data between the 4704 Models 2 and 3 Display Device and the controller.

Coding DKATBL

The DKATBL macro defines composite characters that are to result from the combination of an accent mark and a letter in the same character space. This usage is primarily for national languages that make use of such notation as acute and grave accent marks and the circumflex in French, for example, or the tilde and cedilla in Spanish. All values coded in the macro are hexadecimal representations of the characters as they appear in the user's application program, but without the hex syntactic notation (the X' ... ' is omitted).

These composite characters are distinct entities, each with its own hexadecimal representation (the composites do not result from any sort of combination of the hex values for individual characters). The values used might well vary from country-to-country, and most certainly vary according to the specific letter and accent notation used.

You code the macro with one section for each accent character. Each such accent is followed by one or more letter-and-composite pairs.

Considerations for Changing Over from a 3600 System

The following topics outline some of the major considerations you might encounter.

Storage Allocation

Although the 4700 allocations for the global (GMS), system (SMS), and application program (14) storage segments are larger than for the 3600 Finance Communication System, these larger 4700 allocations do not affect the operation of 3600 programs. The 3600 programs that refer to the GMS and SMS fields defined symbolically with COPY DEFGMS or COPY DEFSMS can run normally. On the other hand, any 3600 programs that use nonstandard symbolic references or define their own GMS or SMS areas might require redefining of those areas. Also, any configurations that are at or near the limits of their segment specifications for the GMS, SMS, or segment 14 might need testing to ensure that enough storage remains to accommodate the larger segments.

Refer to *IBM 4700 Finance Communication System: System Configurator* for information regarding storage requirements and how to determine them.

Unsupported Configuration Macros

The 4700 system does not support the TRMINAL, DEV3618, LDISK, and DCA3289 configuration macros used for 3600 systems.

Selecting Optional Modules

The 3600 OPTMOD configuration macro permits selection of certain optional modules that are now either invalid or an integral part of the 4700 load image.

The following modules are not valid on the 4700, and must be removed before you reconfigure a 3600 load image:

- M18 — 3600 fixed disk error recovery module
- P58 — 3600 fixed disk operation module
- M85 — 3618 printer support module

In addition to these invalid modules, other 3600 RPQ modules (for example, M8D, M91, MA1, MA2) are invalid on the 4700 system. An invalid module causes a configuration error message, which stops the configuration process.

The following modules either have new identifiers or are no longer optional:

- M05 — Backward display operating module
- M0A — Address-sharing support module
- M1C — 3604-7 high-intensity support module
- P20 — LCHAP instruction support module
- P23 — 3270 Data Stream (now part of optional module P70)
- P2D — Expanded LLOAD operation support module

- P60 — Inverse move support module
- P6D — Optional instruction locator

If the 4700 system detects requests for these modules during configuration of a 3600 load image, it issues a warning message and continues with the configuration process.

Defining 3278 Shift Keys

You must now define the 3278 ALT and normal shift keys as transient on the SHFT operand of the TRTBHDR configuration macro.

System Monitor Requirements for Using the Host Transmission Facility

If your 4700 system updates or creates an operating diskette using the 4700 host transmission facility, the STARTGEN macro must specify an expanded CNM/CS version of the system monitor. The standard monitor — the one you get by default — cannot handle these operations.

You must also code the EDAM and ALLOC operands on the FILES macro, and OPTMOD must select modules P40 and P5E. Refer to the descriptions of the STARTGEN and FILES macros, later in this book, for additional clarification about these requirements.

Chapter 2. Examining a Sample CPGEN

This chapter shows the macros for producing a sample configuration. Figures 2-1 and 2-2 show the configuration produced by these macros.

When you develop a configuration, two main considerations apply.

- The *physical* layout and interconnection of the devices in the configuration.
- The use to which the configuration is to be put. We call this the *logical* layout. It is really the arrangement of the devices into *work stations*.

That is why we show two figures. The two distinct aspects to any configuration eliminate any suitable way to portray them as a single thing.

Introducing Configuration Concepts

Figure 2-1 represents the physical layout and organization. It shows a controller and its link to the host computer, and a loop to which four devices attach — two display-keyboard devices and two printers. Note the macro instructions shown in the figure. These relate the code in the example to the various pieces in the overall collection of devices.

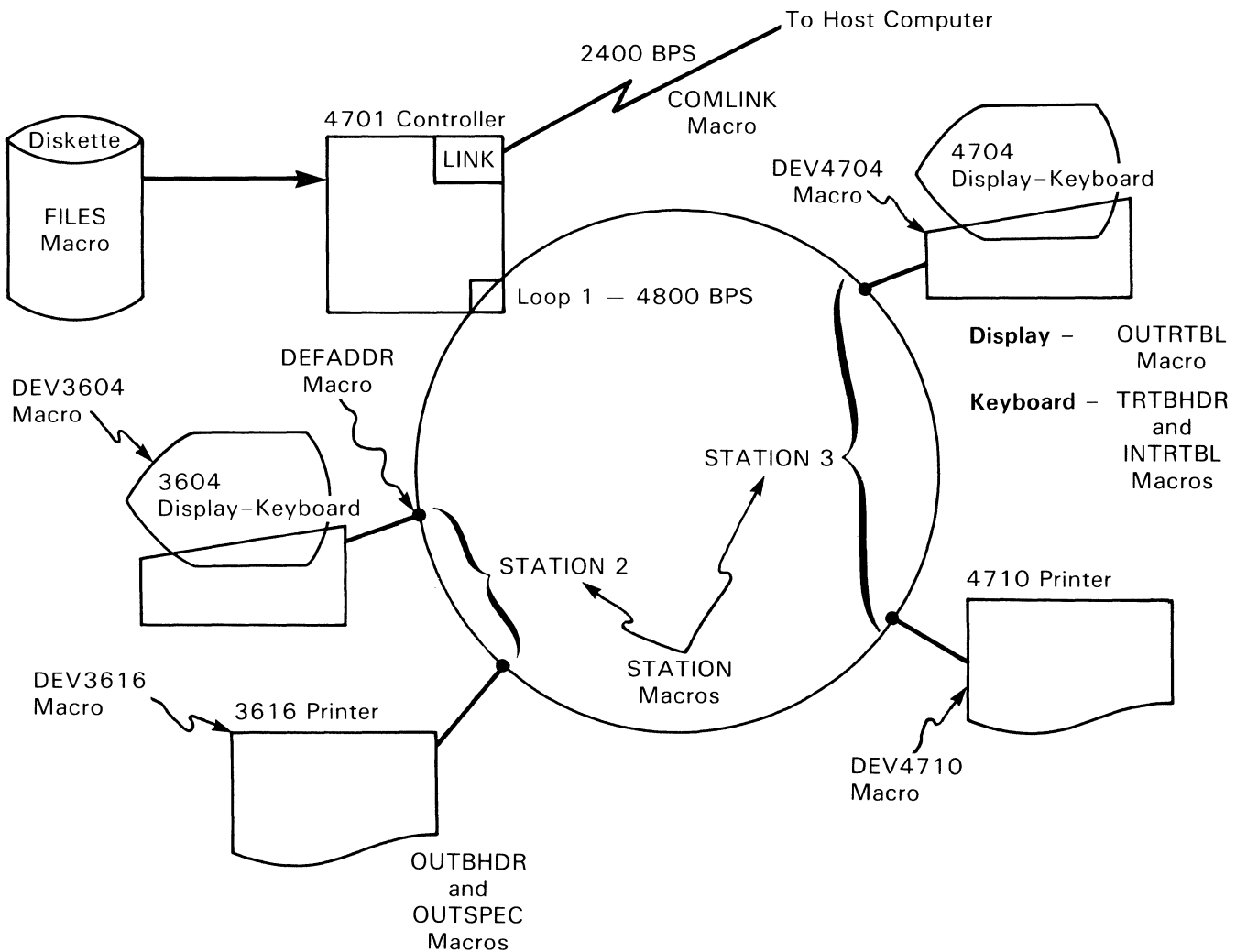
For example, the COMLINK macro establishes and defines the nature of the controller's link to the host. For each device, a DEVxxxx macro defines the device for the configuration in programming terms. (The xxxx is really the numeric part of a device's name, such as 4704 or 4720.)

The DEVxxxx macro also relates other necessary macros — such as those that describe the character-translation tables for the keyboards and printers. If our example included devices attached through a device cluster adapter, we would need to use some DCAxxxx macros for the same purpose. Our example does not use such devices, however.

The DEFADDR macros define the points of attachment of the devices to the loop. They make the devices *addressable* by their controller.

For the configuration to work as a whole, many of these macros have to relate to one another. For example, INTRTBL, OUTRTBL, TRTBHDR, and OUTSPEC macros, where they apply, must be logically associated with their respective devices. To do this, you give the macros required labels. These labels are symbolic names by which other macros in the related group can identify them by using the names in their operands.

Chapter 3 identifies macros that require labels. For each such macro, the description in Chapter 3 states the relationship that results when another macro uses the label and identifies the other macro (or macros) that will use it.



Macros

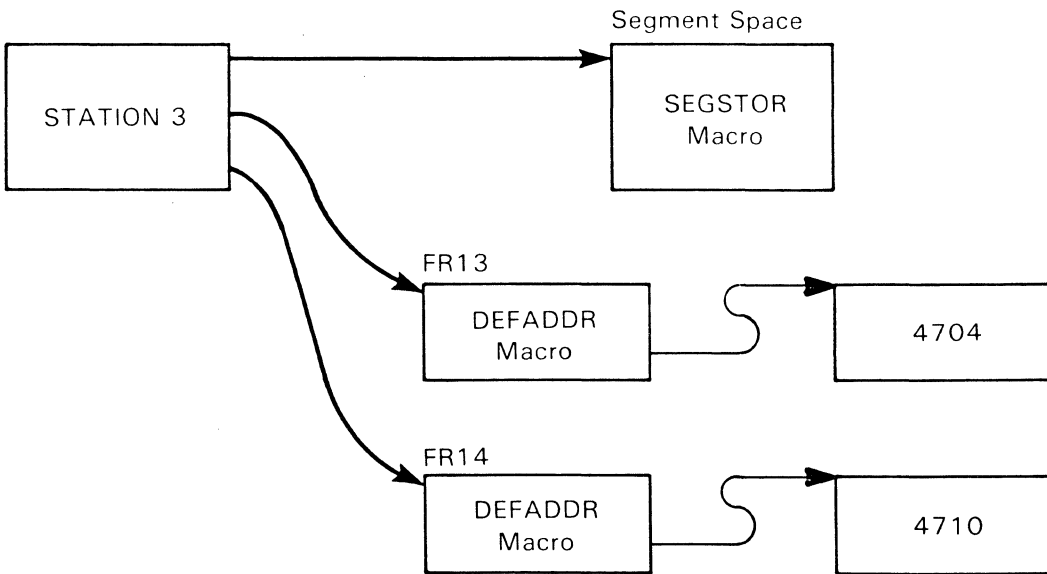
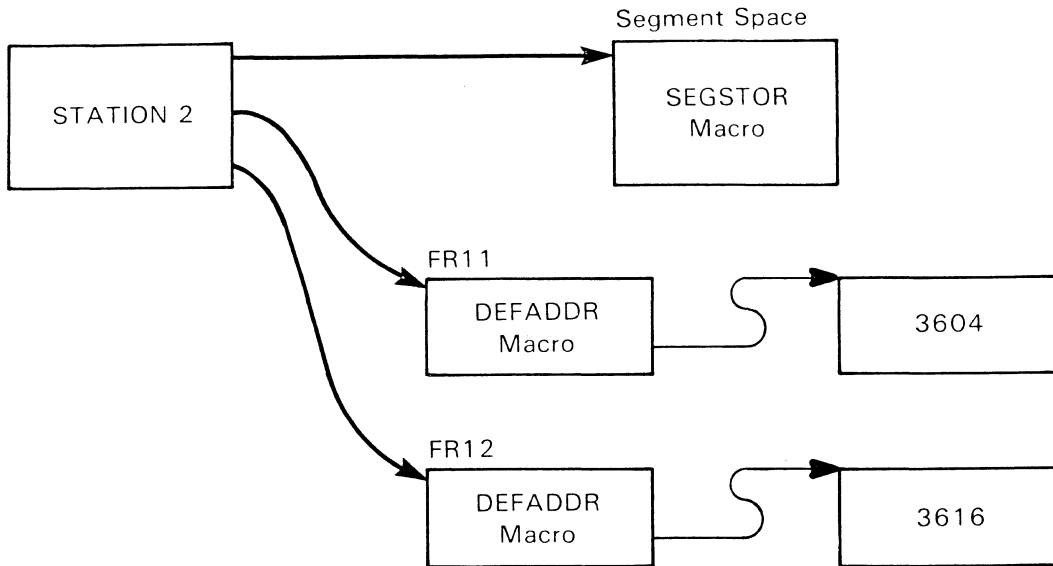
Purpose

FILES	Specifies the diskette.
DEFADDR	Specifies an address on the loop.
DEV3616	Specifies a 3616 printer.
OUTBHDR	Points to an OUTSPEC macro.
OUTSPEC	Specifies the character set that the printer is to use.
DEV4704	Specifies a 4704 display-keyboard device.
OUTRTBL	Specifies the translation table that the display is to use.
TRTBHDR	Points to the INTRTBL macro.
INTRTBL	Specifies the keyboard character set.
DEV3604	Specifies a 3604 display-keyboard device.
DEV4710	Specifies a 4710 printer.
STATION	Defines the work stations on the loop.
COMLINK	Defines the communication link between controller and host.

Figure 2-1. Overall Physical Configuration

Figure 2-2 emphasizes the way the configuration is to be used. Even though it looks very different from Figure 2-1, it really shows the same collection of devices. Here, however, we have assembled the devices as two work stations. Although we didn't have to, we happened to put the two 3600 Finance Communication System devices into the same work station, and paired the 4700 devices into the other.

The important macros for the work station concept are `STATION` and `SEGSTOR`. `STATION` is a macro whose coding can become quite long. It defines the work station by naming the devices that are to be a part of it and stating the capabilities the work station is to have. `SEGSTOR` reserves a discrete segment of controller storage for each such station to use without contention from other work stations.



STATION Macro — Specifies station characteristics, assigns segment space to the station, assigns devices with their associated LDAs to the station.

SEGSTOR Macro — Specifies the number and size of the segments to be assigned to the station.

Figure 2-2. Logical Layout (Configuration) of Work Stations

This example is organized as follows:

- First is the STARTGEN macro. Every CPGEN must begin with a STARTGEN macro.
- Next is the COMLINK macro, which defines the controller's link with its host computer.
- The STATION macros are next, to define the two work stations. These could just as well come later, after macros for defining the physical configuration.
- The SEGSTOR macros happen to be after the STATION macros, to which they relate, and where you might expect to find them.
- The DEFADDR macros — called *terminal descriptions* in the code comments — are next. They define the points of loop-attachment for the devices that we will define next. That gives the devices addressability from the controller.
- The device-defining macros — DEVxxxx macros — are next.
- Finally are all the macros necessary for defining all the character-translation-table information for all the devices. These constitute quite a collection. Each device requires its own set of translation-table macros.

Another way of organizing these would be to associate each set with each device by coding the macros right under their respective DEVxxxx macro. Either order works well. The way you choose to do it depends largely on what you find easiest to work with and understand.

The code in boxes, or following an asterisk (*) in column 1 of each code line, is comments. The comments take no part in defining the configuration. They simply help explain the listing and divide it visually. They relate to the notes of explanation immediately above.

The Xs are continuation indicators. They tell the macro compiler (CPGEN processor) that the information that follows in the next line is logically a part of the same statement. Their column position and coding standards follow the standard conventions for VM/370 assemblers.

The following example produces the configuration in Figures 2-1 and 2-2. First is the STARTGEN macro, which gives historical, or accounting sort of information about the CPGEN. Every CPGEN must begin with one of these.

```

*-----*
*
*           STARTGEN MACRO
*           START CONFIGURATION PROCEDURE
*
*-----*
STARTGEN ID=SAMPL00,      CONFIGURATION ID      X
DATE=010183,             DATE OF LATEST UPDATE X
CTLR=4701-1,            CONTROLLER MODEL      X
S13=0,                  GLOBAL SEGMENT 13      X
S15=0,                  GLOBAL SEGMENT 15      X
VN=0                    VERSION NUMBER
EJECT

```

Next is the COMLINK macro, which establishes the communication link between the host operating system and a controller. The operands of the COMLINK macro define certain specifics for the communication link. Look at the comments ... they reveal the specifics.

```

*-----*
*
*           COMLINK MACRO
*           DEFINE COMMUNICATIONS LINK
*
*-----*
COMLINK DCL=2400,        SPEED OF HOST LINK IN BPS X
CTG=:40,                LINK TIME OUT - MIN:SEC X
CNL=256,               READ BUFFER SIZE      X
CNB=2,                 NUMBER OF READ BUFFERS X
WRT=2,                 NUMBER OF WRITE BUFFERS X
TYPE=4502              SDLC-SNA
EJECT

```

The FILES macro defines diskette files that are to be in use in the configuration. As the comments below indicate, this FILES macro defines a permanent file, a temporary file, and subfiles for the temporary file.

```

*-----*
*
*           FILES MACRO
*           SPECIFY DISKETTE FILES
*
*-----*
FILES PF=7,             PERMANENT FILE - NO. OF BLOCKS X
TF=(1,10),            TEMPORARY FILE - (NUMBER,TRACKS)X
INDX1=(2-7)          SUBFILES FOR TF - (LOW-HIGH)
EJECT

```

```

*-----*
*
*           SPECIFY OPTIONAL MODULE LIST
*
*-----*
*
* NOTE: IF YOU WANT AUTOMATIC WARM START CONTROL
*       TO PREVENT AN ACCIDENTAL COLD START -
*       YOU CAN ADD THE OPTMOD MACRO
*
*       OPTMOD SETDSKT      WARM START CONTROL
*       EJECT
*-----*

```

The LOOPS macro below defines characteristics of the loop to which our sample configuration's devices are to attach. Its operands give the loop an identifying number and type, establish its operating speed, and describe its modem capability.

```

*-----*
*
*                               *
*                               *
*                               *
*                               *
*                               *
*-----*
L1          LOOPS ID=1,          LOOP NUMBER 1          X
            TYPE=L,            LOCAL LOOP              X
            BPS=4800,          SPEED OF LOOP IN BPS     X
            MODEM=N           NONWRAPPABLE MODEM
            EJECT

```

Next are the STATION macros, which detail the way our sample configuration's devices are to be arranged. Our configuration needs two STATION macros. One is for the work station containing the 4700 devices. The other is for the work station containing the 3600 devices. The comments on the code lines for each macro denote how or why the operands are used here.

```

*-----*
*
*                               *
*                               *
*                               *
*                               *
*                               *
*-----*
FASTA02A STATION ID=2,          WORK STATION IDENTIFICATION NO.X
            TERM=(FR11,FR12(3,4,A)), TERMINALS (LOGICAL-ADDRESS)X
            RETSTK=18,          RETURN STACK            X
            SS=1,               STORAGE CLASS           X
            APBNM=(SAMPLE01,1), (APPLICATION-NAME, VERSION-NO.)X
            DELSET=X'0F',       DELIMITER CONTROL BYTE X
            CPU=Y,              COMMUNICATES WITH THE CPU X
            STARTUP=Y,          APPL GETS CONTROL AT STARTUP X
            WRT=1               SEQUENCE CHECK EACH CPU WRITE
            EJECT
FASTA03A STATION ID=3,          WORK STATION IDENTIFICATION NO.X
            TERM=(FR13,FR14(3)), TERMINALS (LOGICAL-ADDRESS) X
            RETSTK=18,          RETURN STACK            X
            SS=1,               STORAGE CLASS           X
            APBNM=(SAMPLE02,1), (APPLICATION-NAME, VERSION-NO.)X
            DELSET=X'0F',       DELIMITER CONTROL BYTE X
            CPU=Y,              COMMUNICATES WITH THE CPU X
            STARTUP=Y,          APPL GETS CONTROL AT STARTUP X
            WRT=1               SEQUENCE CHECK EACH CPU WRITE
            EJECT

```

Next, the SEGSTOR macros define the way the controller storage is to be segmented for operating purposes. Note from the comments that our configuration does not use eight of the possible 13 user-definable storage segments that the SEGSTOR macro can define. Segments 5-12 have zero coded as their operands.

```

*-----*
*
*                SEGSTOR MACROS
*            SPECIFY SEGMENT STORAGE
*
*-----*
          SEGSTOR CLASS=1,      STORAGE CONFIGURATION CLASS      X
              SEGSIZE=( 30,    SEGMENT 0 - OPERATOR/CASHBOX/STATIONX
              300,             SEGMENT 2 - I/O BUFFER AREA        X
              170,             SEGMENT 3 - CUSTOMER      DURATION  X
              250,             SEGMENT 4 - TRANSACTION DURATION  X
              0,               SEGMENT 5 - NOT USED              X
              0,               SEGMENT 6 - NOT USED              X
              0,               SEGMENT 7 - NOT USED              X
              0,               SEGMENT 8 - NOT USED              X
              0,               SEGMENT 9 - NOT USED              X
              0,               SEGMENT 10 - NOT USED             X
              0,               SEGMENT 11 - NOT USED             X
              0),              SEGMENT 12 - NOT USED             X
          S13=800              SEGMENT 13 - SAMPLE GLOBAL DURATION
EJECT

```

The DEFADDR macros each establish addressability for one of our four devices in the configuration. Note that the symbolic names in the operands — FR11A, FR12A, FR13A, and FR14A — are the same as the labels of the DEVxxxx macros that follow in the next group of macros. This technique relates each device description (given in the device's DEVxxxx macro) with its logical address, as defined in its DEFADDR macro.

```

*-----*
*
*                DEFADDR MACROS
*
*            TERMINAL DESCRIPTIONS
*
*            FR11 - 3604-6 45 -CHARACTER KEYBOARD INPUT
*                       175-CHARACTER DISPLAY
*            FR12 - 3616  US CHARACTER PRINTER
*                       175-CHARACTER DISPLAY
*            FR13 - 4704-11 112-CHARACTER KEYBOARD INPUT
*                       175-CHARACTER DISPLAY
*            FR14 - 4710  US CHARACTER PRINTER
*
*-----*
FR11    DEFADDR FR11A,      3604 SMALL KEYBOARD      X
          ADDR=( L1,1,0 ) ( LOOP, TERMINAL-ADDRESS, MODULUS )
*
FR12    DEFADDR ( FR12A,2 ), PASSBOOK AND DOCUMENT PRINTER      X
          ADDR=( L1,2,0 ) ( LOOP, TERMINAL-ADDRESS, MODULUS )
*
FR13    DEFADDR FR13A,      4704 SMALL KEYBOARD      X
          ADDR=( L1,3,0 ) ( LOOP, TERMINAL-ADDRESS, MODULUS )
*
FR14    DEFADDR ( FR14A,2 ), JOURNAL AND DOCUMENT PRINTER      X
          ADDR=( L1,4,0 ) ( LOOP, TERMINAL-ADDRESS, MODULUS )
EJECT

```

The DEVxxxx macros each describe the particular characteristics that one of the generic devices is to have when operating in our sample configuration. As noted in the preceding description of the DEFADDR macros, each DEVxxxx macro has as its label the same symbolic name as used as an operand in one of the DEFADDR macros. This linking of symbolic names couples each of the device macros with a specific one of the DEFADDR macros, thus establishing logical addressability for the device in the configuration.

```

*-----*
*
*                               DEVxxxx MACROS
*
*-----*
* NOTE: IF YOU DO NOT HAVE A MAGNETIC STRIPE ENCODER,
*        YOU CAN DELETE THE OPTIONS=4905 PARAMETER
*
FR11A   DEV3604 MODEL=6,          MODEL NUMBER                X
          OPTIONS=4905,          MAGNETIC STRIPE ENCODER      X
          TRTBHDR=IN3604,        INPUT TRANSLATION TABLE    X
          OUTRTBL=OUT3604        OUTPUT TRANSLATION TABLE
*
FR12A   DEV3616 OUTBHDR=OUT3616, OUTPUT TABLE TRANSLATE HEADER X
          PS=(0,24),             PAGE SIZE ON JOURNAL/DOCUMENT X
          WL=0,                  WARNING LINE                  X
          CFOLD=(12,2),          CENTER FOLD BEGIN, SKIP      X
          OFFSET(1,24),          OFFSET                          X
          SHARED=Y,              SHARED PRINTER                X
          CONSHR=Y,              CONCURRENT JOURNAL PRINTING  X
          KEY=(N,Y),             AUTOSTART JOURNAL/DOCUMENT   X
          NL=AUTO,               AUTO NEW LINE                  X
          PBFOLD=HORZ,           HORIZONTAL FOLD PASSBOOK     X
          MARGIN=0,              LEFT MARGIN PASSBOOK         X
          LPI=5,                  LINES PER INCH                X
          CPI=(1,1),             CHARS PER INCH                X
          LINE=(56,100),         LINE LENGTH JOURNAL/DOCUMENT X
          DEFTRT=USA3616        DEFAULT TRANSLATION TABLE
          EJECT
*
FR13A   DEV4704 MODEL=11,         MODEL NUMBER                X
          OUTRTBL=OUT4704,       OUTPUT TRANSLATION TABLE    X
          OPTIONS=4905,          MAGNETIC STRIPE ENCODER      X
          TRTBHDR=IN4704,        INPUT TRANSLATION TABLE    X
          CHARSET=US             US CHARACTER SET
*
FR14A   DEV4710 OUTBHDR=OUT4710, OUTPUT TABLE TRANSLATE HEADER X
          PS=4,                  PAGE SIZE ON JOURNAL/DOCUMENT X
          WL=0,                  WARNING LINE                  X
          OFFSET=0,              OFFSET                          X
          SHARED=N,              NONSHARED PRINTER            X
          KEY=N,                 AUTOSTART JOURNAL/DOCUMENT   X
          CPI=1,                 CHARACTERS PER INCH = 12     X
          LINE=48,               LINE LENGTH JOURNAL/DOCUMENT X
          DEFTRT=USA4710        DEFAULT TRANSLATION TABLE
*

```

Next, the TRTBHDR macros specify the headers for the translation tables for the keyboards of each of our two display devices. Note how the operands KBTB45L and KBTB45U relate to the same names used as labels for the INTRTBL macros, which follow.

```

*-----*
*
*                               TRTBHDR MACROS
*                               SPECIFY TRANSLATION TABLES
*                               KBTB45 45 CHARACTER KEYBOARD INPUT
*                               US      112 CHARACTER KEYBOARD INPUT
*
*-----*
IN3604  TRTBHDR CANCEL=0E,        RESET/CANCEL KEY            X
          INTRTBL=(KBTB45L,KBTB45U) (LOWERCASE, UPPERCASE)
*
IN4704  TRTBHDR CANCEL=31,        RESET-CANCEL KEY            X
          INTRTBL=UPPER UPPERCASE ONLY
          EJECT

```

The INTRTBL macros define the actual information that is to be in each of the translation tables for our two keyboards. KBTB45L describes the lower-case character set, while KBTB45U describes the upper-case character set.

The INTRTBL macro specifies a keyboard as a feature number or keyboard type. You can also use the macro to modify or define any key on the keyboard.

Each keyboard is unique and has its own set of engraved or non-engraved keys (or both). Refer to *4700 Controller Programming Library, Volume 4* for the keyboard layouts.

Our example shows two extremes in regard to modifying a keyboard:

- The 3604 Model 6 has a 45-key keyboard consisting of three sets of 15 keys each. These keys are primarily non-engraved keys and have been defined by the installation. Their definitions each determine what character a particular key represents to a controller application program.
- The 4704 Model 11 has a 112-key US character set, of which the user intends to use only the uppercase definitions. The user has redefined the lowercase keys for the numerals 0-9 also as uppercase.

```

*-----*
*                               INTRTBL MACROS                               *
*                               BUILD INPUT TRANSLATION TABLES             *
*                               KBTB45L 45 CHARACTER KEYBOARD - LOWERCASE    *
*                               KBTB45U 45 CHARACTER KEYBOARD - UPPERCASE    *
*                               UPPER 112 ALL UPPER CASE KEYBOARD          *
*-----*
KBTB45L  INTRTBL (00,X'ED405C5CEE'),          SIGNAL KEY          X
          (01,X'EDD5E4D4EE'),                NUM                X
          (02,X'EDC5D6C3EE'),                EOC                X
          (03,BKSP),                          BACKSPACE          X
          (04,C'/' ),                          /                  X
          (05,C'*' ),                          *                  X
          (08,C'TLRA',EOMA,X'80'),            TELLER A          X
          (09,X'EDD4D6405BEE'),              MO $               X
          (0A,X'EDD4D6EE'),                  MO                 X
          (0B,C'7' ),                          7                  X
          (0C,C'8' ),                          8                  X
          (0D,C'9' ),                          9                  X
          (0E,C'0' ),                          0                  X
          (10,C'TLRB',EOMB,X'80'),            TELLER B          X
          (11,X'EDE2E4C2EE',EOM,X'80'),      SUB                X
          (12,X'EDE4D7C4E3EE'),              UPDT               X
          (13,C'4' ),                          4                  X
          (14,C'5' ),                          5                  X
          (15,C'6' ),                          6                  X
          (16,CAS2),                          CASE 2             X
          (17,C'.' ),                          .                  X
          (18,X'EDD140C4D9EE'),              J DR               X
          (19,X'EDD140C3D9EE'),              J CR               X
          (1A,X'EDC1C4C4EE',EOM,X'80'),      ADD                X
          (1B,C'1' ),                          1                  X
          (1C,C'2' ),                          2                  X
          (1D,C'3' ),                          3                  X
          (1E,X'EDE3D6E3D3EE,EOM,X'80'),     TOTL               X
          (1F,C'00' ),                        00                 X
          (20,X'EDC4D9EE'),                  DR                 X
          (21,X'EDC3D9EE'),                  CR                 X
          (22,X'EDD4C9E2C3EE'),              MISC               X
          (28,X'EDD3D6C1D5EE'),              LOAN               X
          (29,X'EDD7D9C9D9EE'),              PRIN               X
          (2A,X'EDC5E2C3D9EE'),              ESCR               X
          (2B,X'EDC5D5E3D9EE,EOM,X'80'),     ENTR               X
          (30,X'EDE2C1E5EE'),                SAV                X
          (31,X'EDC2C1D3EE'),                BAL                X
          (32,X'EDD3C9D5C5EE'),              LINE               X
          (33,X'EDD3D6C3D3EE'),              LOCL               X
          (38,X'EDC4C4C1EE'),                DDA                X
          (39,X'EDC3C8D2EE'),                CHK                X
          (3A,X'EDC3C1E2C8EE'),              CASH               X
          (3B,X'EDC6C7D5EE'),                FGN                X
          KYBD=4663,CASE=1                    LOWERCASE          X
EJECT

```



```

KBTB45U  INTRTBL (00,C,'A'),          A          X
          (01,C'B'),                  B          X
          (02,C'C'),                  C          X
          (03,BKSP),                  BACKSPACE  X
          (04,ADVANCE),                ADVANCE    X
          (05,X'40'),                  SPACE     X
          (08,C'D'),                  D          X
          (09,C'E'),                  E          X
          (0A,C'F'),                  F          X
          (0B,C'7'),                  7         X
          (0C,C'8'),                  8         X
          (0D,C'9'),                  9         X
          (0F,C'0'),                  0         X
          (10,C'G'),                  G          X
          (11,C'H'),                  H          X
          (12,C'I'),                  I          X
          (13,C'4'),                  4         X
          (14,C'5'),                  5         X
          (15,C'6'),                  6         X
          (16,CAS1),                  CASE 1    X
          (17,C'.'),                  .         X
          (18,C'J'),                  J          X
          (19,C'K'),                  K          X
          (1A,C'L'),                  L          X
          (1B,C'1'),                  1         X
          (1C,C'2'),                  2         X
          (1D,C'3'),                  3         X
          (1E,X'M'),                  M          X
          (1F,C'00'),                 00        X
          (20,C'N'),                  N          X
          (21,C'O'),                  O          X
          (22,C'P'),                  P          X
          (28,C'Q'),                  Q          X
          (29,C'R'),                  R          X
          (2A,C'S'),                  S          X
          (2B,C'Z'),                  Z          X
          (30,C'T'),                  T          X
          (31,C'U'),                  U          X
          (32,C'V'),                  V          X
          (33,X'EDE2C5D5C4EE',EOM,X'80'), SEND      X
          (38,C'W'),                  W          X
          (39,C'X'),                  X          X
          (3A,C'Y'),                  Y          X
          (3B,X'EDC3C9C6EE',EOM,X'80'), CIF      X
          KYBD=4663,CASE=2          UPPERCASE

```

EJECT

*

```

UPPER    INTRTBL (18,C'1'),          UPPERCASE NUMBER 1  X
          (19,C'2'),                  2         X
          (1A,C'3'),                  3         X
          (10,C'4'),                  4         X
          (11,C'5'),                  5         X
          (12,C'6'),                  6         X
          (08,C'7'),                  7         X
          (09,C'8'),                  8         X
          (0A,C'9'),                  9         X
          (0F,C'0'),                  0         X
          KYBD=47US112,CASE=2

```

EJECT

The OUTBHDR, OUTSPEC, and OUTRTBL macros are analogous to the TRTBHDR and INTRTBL macros. TRTBHDR and INTRTBL are translation-table macros for input devices ... keyboards, in our example. OUTBHDR, OUTSPEC and OUTRTBL are for output devices ... printers and display devices in our example.

OUTBHDR establishes a header for a device that is to have an output translation table.

OUTSPEC specifies the data that is to be in the translation table for a printer. The table is for a World Trade National Use difference character set. The OUTSPEC macros in our example merely identify a standard, IBM-supplied US character set.

OUTRTBL specifies the data that is to be in the translation table for a display device. The OUTRTBL macros in our example also specify a standard output character set.

You can use OUTSPEC and OUTRTBL to make rather elaborate changes to the output character sets, however.

```

*-----*
*
*                               OUTBHDR MACROS AND
*                               OUTSPEC MACROS
*
*                               SPECIFY OUTPUT TRANSLATE HEADERS
*
*                               OUT3616  3616  HEADER TABLE
*                               OUT4710  4710  HEADER TABLE
*-----*
OUT3616  OUTBHDR  OUTSPEC=USA3616
OUT4710  OUTBHDR  OUTSPEC=USA4710
USA3616  OUTSPEC  DEVICE=3616,CHARSET=US
USA4710  OUTSPEC  DEVICE=4710,CHARSET=US
                EJECT

*-----*
*
*                               OUTRTBL MACROS
*
*                               BUILD OUTPUT TRANSLATION TABLES
*
*                               OUT3604  175  CHARACTER DISPLAY
*                               OUT4704  175  CHARACTER DISPLAY
*-----*
                SPACE 1
OUT3604  OUTRTBL (39,X'ED'),(32,X'EE'),(32,X'FF'),(32,X'EF'),      X
                DEVICE=3604,          3604 DISPLAY TRANSLATE TABLE X
                CHARSET=175,          STANDARD 175 CHARACTER SET   X
                DEF=32                 WITH BLANK AS THE DEFAULT
                SPACE 2
OUT4704  OUTRTBL
                DEVICE=4704,          4704 DISPLAY TRANSLATE TABLE X
                CHARSET=175,          STANDARD 175 CHARACTER SET   X
                EJECT

```

The ENDGEN macro provides a conclusion to the configuration procedure. ENDGEN also allows you to initialize segments, as indicated in the example. Also, the SEGINIT macro initializes the segments of controller storage on behalf of the user application programs that are to use them.

```

-----*
*
*                               ENDGEN MACRO AND
*                               SEGINIT MACROS
*
*                               END CONFIGURATION PROCEDURE AND
*                               INITIALIZE THE USER APPLICATION PROGRAM
*
-----*
*
*   ENDGEN ( 2,3,84,X'3C13' ),   CONFIG AND CONFIG1 FOR STATION 2 X
*   ( 3,3,84,X'1A00' ),         CONFIG AND CONFIG1 FOR STATION 3 X
*   ( 2,3,88,X'06' ),           NUMBER LINES ON DS FOR STATION 2 X
*   ( 3,3,88,X'0C' ),           NUMBER LINES ON DS FOR STATION 3 X
*   ( 3,3,89,X'41' ),           SETJPTO TO CUTFORM 1 FOR 4710
*
*   SEGINIT ( 1,13,36,X'09' )   GLOBAL THRESHOLD
*   END

```

Chapter 3. Using the Configuration Macros

Controller configuration macro instructions specify the physical and logical configuration of the controller and its associated terminals. When the configuration process takes place, the parameters specified in the macro instructions tailor the controller load image to fit the hardware configuration and desired programming configuration.

The macro instructions that are primarily concerned with the hardware configuration are:

COMLINK
DCAPORT
DEFADDR
DEVnnnn
DCAnnnn
DEVnnnn
EXTCTR
LDA
LOOPS
STARTGEN

The macro instructions that are primarily concerned with the programming configuration are:

APLIST
| APMEDIA
CHARDEF
DCATRTBL
DEFSBF
DEFSPA
DKATBL
ENDGEN
FILES
INTRTBL
MSTRTBL
| OPTLIST
OPTMOD
OUTBHDR
OUTRTBL
OUTSPEC
PINTBL
PRIDSP
PSSLST
SEGINIT
SEGSTOR
STATION
TRANPL
TRTBHDR
TRTLIST

Interpreting the Syntax Notation

A uniform notation describes the syntax of the CPGEN instructions. The notation indicates which operands you must code and which are optional, the options that are available for expressing values, the values assumed by the system if you do not code an operand, and the punctuation. A complete description of how to interpret the symbols and conventions used in this form of notation is in *4700 Controller Programming Library — Volume 1*.

APLIST -- Add Application Programs to Configuration List

APLIST adds application programs (APs) to the AP configuration list. A program defined by APLIST is called by a main application program defined by the STATION macro, but is not assigned to a particular work station. You can specify as many as 255 application programs with one APLIST macro. If you need more, code another APLIST macro. You can specify a maximum of 1024 names per configuration.

Note: APLIST is not supported by LCF, which provides a transfer function for application programs instead.

Name	Operation	Operand
[label]	APLIST	(apname1, [vn 1] [,T R]) [, (apname2, [vn 1] [,T R])] . . .

label

Optional name (1-8 alphameric characters) by which other macros can refer to this macro specification.

apname

Is the program name specified by the APBNM operand of the program's BEGIN instruction. For COBOL programs, *apname* is the program name specified in the PROGRAM-ID paragraph.

vn

Is the version number of the program assembly, a decimal integer in the range 0-99; 1 is the default.

Note: Multiple versions of the same application program are not permissible in one configuration.

T or R

Optional positional operand that enables you to specify whether your called application programs are to be transient or resident. The default value is R, which means that the application program is to be resident in storage. T means that the application program is to be transient; the program's name is included in the directory table of application programs, but the program itself is not loaded during IPL.

| APMEDIA -- Specify Application Program Residence

APMEDIA enables you to specify where (on which disk or diskette drives) your application program data set (SYSAP) and the system monitor's application program data set (SYSSM) are to reside.

LCF does not support the APMEDIA macro.

Name	Operation	Operand
------	-----------	---------

```
[label] APMEDIA [MON={1|A|B}]  
                [,UAP=( [{1|A|B} , {SYSAP|SYSAPxxx} ) ] ]
```

MON

Specifies where the system monitor's application program — SYSSM — is to reside. Residence can be only on a primary diskette drive, which you denote by specifying 1 for the IML diskette, or A or B for a disk drive. The default value is 1.

UAP

Specifies where the user application program — SYSAP or SYSAPxxx — is to reside. The value for xxx can be any combination of characters that uniquely defines a user data set other than the default data set. Residence can be only on a primary diskette drive, which you denote by specifying 1 for the IML diskette, or A or B for a disk drive. The default value for the drive specification is 1, and for the application program specification is SYSAP.

Note: If either data set resides on disk, you must include a FILES macro in your CPGEN with the DSK operand specified. Failure to do so will result in an X088 error during startup.

X'a.....s'

Each entry you specify defines one special character. The *a* position represents the first column — or vertical “slice” of the character. The *g* position represents the last slice for a 3616 or 4710. The *i* position represents the last slice for a 4720 printing in its regular mode. The *s* position represents the last slice for a 4720 printing in its quality mode.

The number of hexadecimal values for each column and the number of columns vary as indicated:

X'aabbccddeeffgg' — 3616 or 4710

X'aabbccddeeffgghii' — 4720 regular printing

X'aaaabbbbccccddd....sss' — 4720 quality printing

Code all 16 entries.

Device

Code 3616, 4710, or 4720 for the appropriate printer. If you code 4720, you have a choice between its regular or quality printing modes (code R or Q respectively ... R is the default).

COMLINK -- Define Communication Link

COMLINK describes the link between the 4700 controller and the telecommunication network. You must describe both the modem and the communication buffers.

You can use only one COMLINK instruction in each controller configuration procedure. Default values describe the link if you do not specify the instruction. The COMLINK instruction neither refers to, nor is it referred to by, any other instruction.

Name	Operation	Operand
[label]	COMLINK	[DCL={nnnn 9600}] [,WRAP={Y N}] [,CNL={nnn 72}] [,CNB={n 1}] [,ACB=([DCE] [,DSL] [,PRT] [,SWM])] [,WRT={n 1}] [,CTG=[{mm 0}] [: {0 ss 20}]] [,TYPE=([4502] [,4850] [,1422] [,5656])] [,VERSION={A B}] [,FLAG={AA AC DC}] [,RETRIES={n 5}] [,DEL1={ss.t 03.0}] [,DEL2={ [mm] [:ss] 03:00}] [,SELSEQ=' {c1} [...cn] ' [,OPTIONS=(option[,...]) [,BSC=({X'gpc' C'gpc' X'F0'} , [{m 256}] [, {n 5}] [,DEF])] [,BSCOPT=(option[,...])]

label

Optional name (1-8 alphameric characters) by which other macros can refer to this macro specification.

DCL

Is a decimal number that specifies the link's operating speed in bits per second. The default value is 9600. See TYPE for link speed options.

WRAP

Specifies whether the modem has wrap capability (Y) or not (N). The default value, or the value if TYPE=4850, is N.

CNL

Is the decimal number that specifies the size of the controller's read buffers in bytes. The read buffers must be as large as the largest message read from the communication link — at least 72 bytes.

If you intend to use the operating diskette for the diskette-creating operation, the number of buffers times the size must be at least 256 bytes.

The default value is 72. For BSC3 communication, the read buffers should be as long as the longest message transmitted unless you specify BSCOPT=DRBC.

CNB

Is the decimal number that specifies the number of read buffers. The default value is 1.

ACB

Specifies the characteristics that are valid for the link:

DSL

Indicates that the modem is not normally connected to the line, and you want to connect the modem to the line during STRLINK. The default characteristic is that the modem is normally connected to the line, and the data terminal is ready.

DCE

Indicates that data link communications are done with external modems that use NRZ data encoding. The default value is for NRZI data encoding. Always specify DCE for a BSC3 link that has external modems.

Note: The primary station (3704 or 3705) port and the controllers connected to that port must be consistent in their specification of transmission encoding.

PRT

Indicates that you want a permanent request to send. The default value is for a controlled request to send.

SWM

Specifies that the controller is to be operated on a switched link. The default is for a nonswitched link. If you code this parameter, PRT is ignored. The Start Link (STRLNK) instruction, which also specifies link operating characteristics, does not override the VERSION, RETRIES, DEL1, or DEL2 operands established for this link.

Notes:

1. The VTAM BLKID for the 4700 system host link is X'57'.
2. The omission of tone generation is automatically considered a valid characteristic of the link at the time of the controller configuration procedure; you cannot change it at that time.

WRT

Is a decimal number that specifies the maximum number of outstanding transmissions to the host processor (an outstanding transmission is an LWRITE operation that has not yet received a response from the communications controller). This operand must be greater than or equal to the number (maximum of 7) specified in the WRT operand of any STATION macro instruction. The default value is 1.

CTG

Is a decimal number or numbers indicating the time in minutes (*mm*) and seconds (*ss*) for link time-outs. The total time you specify must not exceed 85 minutes (5100 seconds). You should specify seconds (*ss*) in multiples of 20; if the seconds are not a multiple of 20, the system uses the next lower multiple of 20 and ignores the remainder.

Note: The actual length of the time-out can vary by $\pm 20\%$ of the multiple of 20 you use. If you specify the time only in seconds, be sure to precede the value with a colon (CTG=:*ss*); you can also precede the colon with a 0 to indicate no minutes (CTG=0:*ss*). If you specify the time only in minutes, you omit the colon, and the system assumes 0 for *ss*. If you do not specify CTG, the system assumes one 20-second time-out. If TYPE=4850, specify *ss* in multiples of 8 seconds.

TYPE

Specifies the types of communication link load modules to be included on the diskette. If you do not specify TYPE, the system includes a 4502 on the diskette. You must specify all desired modules in the operand. Available modules are as follows:

4502

Indicates a communication link load module that supports SDLC-SNA.

1422

Specifies a transmission protocol of BSC and sets default values for all other BSC operands. DCL must specify 4800 or less.

4850

Specifies the multiple-use communication adapter for attachment to 8100 systems.

5656

Indicates a communication link load module that supports an X.21 9600 bps switched interface.

VERSION

Specifies the type of diagnostic support to be included for an X.21 link:

A

Selects link diagnostics without tests for the X.21 controller feature. This option is the default.

B

Selects link diagnostics that include tests for the X.21 controller feature. This option requires an addition of 1.0 to the controller storage attachment factor.

FLAG

Specifies the calling and answering options for the X.21 link. The application program can change these options using the Start Link (STRLNK) instruction:

AA

Selects Autoanswer operation for incoming calls. This is the default value.

AC

Selects Autocall operation with an address.

DC

Selects Autocall operation using Direct Call capability.

Note: Authorities can require that the following parameters be specified according to network regulations.

RETRIES

Specifies how many times the controller retries an Autocall operation before performing a log operation. The controller retries calls only if it receives call progress signals. You can specify any number in the range 0-255 for *n*. The default value is 5, which causes an Autocall operation to be performed six times.

DEL1

Specifies the time in seconds and tenths of a second that the controller waits between Autocall retry attempts. Seconds (*ss*) can be any value in the range 00-25. Tenths of a second can be any value in the range 0.0-0.5. The default value is 3.0 seconds.

DEL2

Specifies the time in minutes and seconds that the controller waits between groups of retry attempts specified by the RETRIES operand. The controller continues to retry indefinitely, using the DEL2 value, as long as it receives call progress signals or until the execution of a Stop Link (STPLNK) instruction.

Minutes and seconds can be any value in the range 00-59. You can specify minutes and seconds separately, but you should precede seconds with a colon (for example, :30). The default value is three minutes (03:00).

SELSEQ

Specifies a network-dependent selection sequence comprising as many as 24 EBCDIC characters (including delimiter characters). The permissible EBCDIC delimiter characters are: , (comma), / (slash), - (minus), . (point or period), and the ending delimiter of + (plus).

Note: The characters in the selection sequence must be within single quotation marks.

OPTIONS

Specifies the SNA characteristics to be assumed:

NOEXP

Specifies that messages received for a secondary logical unit (SLU) are to appear in the order received by the controller, rather than as defined by SNA. Under SNA, expedited-flow messages appear before normal-flow messages. For some applications, this SNA-defined method of message handling might pose a problem for programs that handle messages sequentially. The NOEXP specification is thus useful for such programs because it ensures the sequential presentation of messages.

PACE

Specifies that inbound pacing is to be used. If you specify PACE, and the BIND indicates that pacing is active, an LWRITE CP instruction will complete with status but without data transfer taking place if either of the following conditions is pending:

- Data sent from the host is present but not yet read.
- A fast definite response must be sent.

BIND

Specifies that the SNA bind parameters are to be passed to the secondary logical unit (SLU). An LREAD CP must be issued to get the parameters.

NOPURGE

Specifies that negative responses sent by the SLU resulting from any part of a chained message do not cause the remainder of the message to be purged. Instead, the remaining portions of the message are presented to the SLU. The default is that negative responses **do** cause the remaining portion of the chained message to be purged.

LUASSIGN

Specifies that the optional functions of the Assign instruction are to be included. If you specify this option, a positive response is issued to an ACTLU or DACTLU for LU addresses 2 — 60. If you do **not** specify this option, a negative response with sense information is issued to all requests not associated with a station.

LUSENSE

Specifies that a negative response with sense X'80XX' on the LU-LU flow will be presented to the application program. The LU-LU session ends.

The following operands are for a BSC3 communication link only:

BSC

Specifies that the following BSC controller default parameters are to be changed.

gpc

Specifies the one-byte or two-byte general poll address for the controller. X'F0' is the default.

m

Specifies the maximum length of the data in a batch transmission including BSC control characters. The default is 256. The maximum is 4095.

n

Specifies the maximum number of messages to be included in a batch transmission. The default is 5. The maximum is 60.

DEF

Specifies that the default for starting the communication link is batch transmission mode. The link starts in single message mode if you do not specify this parameter, which can be overridden by the control operator or by the controller application program using the STRLNK instruction.

BSCOPT

Specifies that one or more of the following BSC options are to be provided:

DRBC

Specifies that a message whose data portion is greater than the controller's input buffer size is to be passed to the program intact, rather than truncated. This means that incoming messages that are too long do not have to be segmented by the host application program.

DSAD

Specifies that the controller is to verify that all two-byte station addresses are duplicate addresses, and that the controller is not to respond to the selection sequence if the two address bytes do not match. (If DSAD is **not** specified, and the station address is invalid, the controller responds to the selection sequence, and then passes the subsequent message to a default work station.)

ADDR

Specifies that the 3270 poll-select addressing is to be used. Bit 2 of the poll address is set to 1 to derive the select address. An X'C1' poll address becomes the X'E1' select address. Valid poll addresses are X'C1'- X'C9' and X'D1'-D9'.

WACK

Specifies that a WACK response should be sent to a SELECT if no read buffer is available. The normal response is NAK.

CONV

Specifies that data can be returned from the host in response to data transmitted from the controller.

DCAPORT -- Specify a Device Cluster Adapter Port Assignment

The DCAPORT macro associates a DCA device with a DCA port.

Name	Operation	Operand
label	DCAPORT	(labeldca[,poolname]),PORT=addr

label

Is a name (1-8 alphameric characters) of this port definition, and is used in the DA operand of the STATION macro.

labeldca

Is the name (1-8 alphameric characters) of a DCA4704, DCA3262, DCA3278, DCA3279, DCA3287, or DCA5210 macro that defines the terminal assigned to this port.

poolname

A symbolic name (1-8 alphameric characters) that matches the name of the device pool on the DPOOL operand of the STATION macro.

PORT

Is the port address, *addr*, in the range 0-7.

DCATRTBL -- Define a DCA Translation Table

The DCATRTBL macro defines a translation table for a device attached through the device cluster adapter. You use this macro only if you intend to use the 4704-2 device in its local-keytracking mode.

Name	Operation	Operand
label	DCATRTBL	[({scancode,X'n'})[,...]] TYPE={ALT SHIFT}, DEVICE=4704, CASE={1 2 3 4}, IND={Y N}

label

Is the required symbolic name (1-8 alphameric characters) that the TRTLIST macro instruction uses to identify this macro.

scancode

Is the key scan code of the key to be altered. The scan codes are hexadecimal numbers that you code without the X and single-quotation-mark descriptors.

X'n'

Is the one-byte internal code to be associated with the scan code specified. The code represents either the character code to be tracked to the display when the key is pressed, or X'FF' to indicate an attention-generation key such as the Enter key.

Note: Any key that has a corresponding display code of X'FF' causes the key scan code to be presented to the subsystem. This scan code is used to locate the corresponding EID as defined in the INTRTBL macro for the particular case. Any key having a corresponding display code of X'C0' through X'DF' represents internal function keys as described in *4700 Controller Programming Library, Volume 4*. All remaining display codes represent either the character to be tracked or a no-operation code (X'BF').

ALT

Specifies that the table to be generated is to represent the relationship between the Alt key and keyboard scan code and the device regen character code.

SHIFT

Specifies that the table to be generated is to represent the relationship between the keyboard scan code and the device regen character code. You use the CASE operand in conjunction with the SHIFT operand to determine the keyboard shift case.

DEVICE

Always code 4704 with this operand.

CASE

Specifies the shift case to be used in conjunction with the SHIFT operand. The default value is Case 1.

IND

Specifies whether the shift indicator, which represents the keyboard's current shift case, is to be displayed in the indicator row. The default value is *N*.

DCA3262 -- Define a 3262 Printer

The DCA3262 macro, with a DCAPORT macro, defines a 3262 printer attached through the device cluster adapter.

Name	Operation	Operand
label	DCA3262	[PS={n 0}] [,TM={n 0}] [,SWITCH={Y N}] [,CHAIN={Y N}] [,UDDPC1={value 00}]

label

Is the required name (1-8 alphameric characters) for this device description, used in a DCAPORT macro.

PS

Is a decimal number that specifies the number of lines printed on a page. The valid range is 0-255. If you specify zero, no page size is used. The default page size is zero. Any SVF (Set Vertical Format) control code in the data stream can change the page size.

TM

Is a decimal number that specifies the number of lines to be skipped for a top margin before the first printed line on a page. The valid range is 0-255. If you specify zero, no skip is performed. The default margin is zero. (Note that this value is the default margin; any SVF control code in the data stream can change this value.)

SWITCH

Specifies whether activation of the six or eight lines-per-inch, single-space or double-space, and mono-case or dual-case switches is reported as status in SMSDST. If you omit this operand, status is not reported.

CHAIN

Specifies whether data chaining is to be active when printing at the 3262. Y allows data chaining, which is necessary to operate the 3262 at its highest rated printing speed. N, the default, does not permit data chaining. You can also specify chaining dynamically using the DEVARM instruction.

UDDPC1

Optional operand that identifies the device component (printer component) to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF. Note that you do not code an X (to denote hexadecimal) or the usual single quotation marks used with the X in normal notation. The default value is 00.

DCA3278 -- Define a 3278 Keyboard Display

The DCA3278 macro, along with a DCAPORT macro, describes a 3278 keyboard display station attached through the device cluster adapter.

Name	Operation	Operand
label	DCA3278	MODEL=8,OUTRTBL=labelout [,TRTBHDR=labeltrt] [,EOMSET={X'xx' X'FF'}] [,CURSOR={Y N R}] [,MSTRTBL=msintrt] [,ERTLS={Y N}] [,OPTIONS=(option[,...])] [,UDDccc={value 00}] [,FEATURE={MSR APL}]

label

Is the symbolic name (1-8 alphanumeric characters) used to refer to this device description from a DCAPORT macro.

MODEL

Code 8 for a 3278 Model 2.

OUTRTBL

Specifies the label of the OUTRTBL macro that describes the output translation table to be used with the display screen.

TRTBHDR

Specifies the label of the TRTBHDR macro that describes the input translation table header to be used with the keyboard. If this is omitted, the universal translation table is used.

EOMSET

Specifies a one-byte hexadecimal number indicating the value of the EOM set-selection mask for the 3278 keyboard. X'FF' is the default value.

MSTRTBL

Denotes the MSTRTBL macro that defines the input translation table for the magnetic stripe reader, where *msintrt* is the label of that macro.

CURSOR

Specifies how to display the cursor:

- Y — Cursor remains on from the first I/O operation until power is switched off.
- N — Cursor is not used.
- R — Cursor remains on the screen only while the application program is reading from the terminal.

R is the default value for this operand.

ERTLS

Specifies whether:

- Y — An LREAD is posted complete without status as soon as the input segment is full.
- N — Wrong-length status is indicated when the next character is entered *after* the input segment is full. This is the default value.

OPTIONS

Specifies the 3278 options:

- NSML Specifies that this device is not to be used to log onto the system monitor (“No System Monitor Logon”).
- NSCC After you press the Cancel key, the keyboard remains in the current shift case. If you omit it, the keyboard returns to Case 1 after you press Cancel.

UDDccc

Optional operand (or specification of multiple operands) that identifies the device component (or components) to which the user-defined data is to apply. The *ccc* represents a two-character or three-character qualifier that generically describes the device, as follows:

- UDDKB — Keyboard
- UDDDS — Display

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device component in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

FEATURE

Specifies the features that can attach to the 3278-2:

- MSR Indicates that the magnetic stripe reader (a special feature) is to be attached. If you omit this specification, the configuration cannot support this special feature.
- APL You must code this option to be able to use APL. It specifies that the terminal is to use extended data stream support to support APL characters. This support is known as data stream mapping —DATSM — refer to *Controller Programming Library, Volume 4* for additional information about DATSM.

DCA3279 -- Define a 3279 Keyboard Display

The DCA3279 macro, along with a DCAPORT macro, describes a 3279 keyboard display station attached through the device cluster adapter.

Name	Operation	Operand
label	DCA3279	MODEL={8A 8B},OUTRTBL=labelout [,TRTBHDR=labeltrt] [,EOMSET={X'xx' X'FF'}] [,CURSOR={Y N R}] [,MSTRTBL=msintrt] [,ERTLS={Y N}] [,OPTIONS=(option[,...])] [,UDDccc={value 00}] [,FEATURE=APL]

label

Is the symbolic name (1-8 alphameric characters) used to refer to this device description from a DCAPORT macro.

MODEL

Code either:

- 8A for a 3279 Model 2A (four-color) device

- OR -

- 8B for a 3279 Model 2B (seven-color) device.

OUTRTBL

Specifies the label of the OUTRTBL macro that describes the output translation table to be used with the display screen.

TRTBHDR

Specifies the label of the TRTBHDR macro that describes the input translation table header to be used with the keyboard. If this is omitted, the universal translation table is used.

EOMSET

Specifies a one-byte hexadecimal number indicating the value of the EOM set-selection mask for the 3279 keyboard. X'FF' is the default value.

CURSOR

Specifies how to display the cursor:

- Y — Cursor remains on from the first I/O operation until power is switched off.
- N — Cursor is not used.
- R — Cursor remains on the screen only while the application program is reading from the terminal.

R is the default value for this operand.

MSTRTBL

Denotes the MSTRTBL macro that defines the input translation table for the magnetic stripe reader, where *msintrt* is the label of that macro.

ERTLS

Specifies whether:

- Y — An LREAD is posted complete without status as soon as the input segment is full.
- N — Wrong-length status is indicated when the next character is entered *after* the input segment is full. This is the default.

OPTIONS

Specifies the 3279 options:

- NSML Specifies that this device is not to be used to log on to the system monitor (“No System Monitor Logon”).
- NSCC After you press the Cancel key, the keyboard remains in the current shift case. If you omit it, the keyboard returns to Case 1 after you press Cancel.

UDDccc

Optional operand (or specification of multiple operands) that identifies the device component (or components) to which the user-defined data is to apply. The *ccc* represents a two-character or three-character qualifier that generically describes the device, as follows:

- UDDKB — Keyboard
- UDDDS — Display

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device component in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

FEATURE

Specifies that the APL feature is to be included in the configuration for the 3279:

APL

You must code this option to be able to use APL. It specifies that the terminal is to use extended data stream support to support APL characters. This support is known as data stream mapping —DATSM — refer to *Controller Programming Library, Volume 4* for additional information about DATSM.

DCA3287 -- Define a 3287 Printer

The DCA3287 macro, with a DCAPORT macro, defines a 3287 printer attached through the device cluster adapter.

Name	Operation	Operand
label	DCA3287	[PS={n 0}] [,TM={n 0}] [,SWITCH={Y N}] [,UDDPC1={value 00}]

label

Is the name (1-8 alphameric characters) for this device description, and is used in a DCAPORT macro.

PS

Is a decimal number that specifies the number of lines printed on a page. The valid range is 0-255. If zero is specified, no page size is used. The default page size is zero. Any SVF (Set Vertical Format) control code in the data stream can change the page size.

TM

Is a decimal number that specifies the number of lines to be skipped for a top margin before the first printed line on a page. The valid range is 0-255. If you specify zero, no skip is performed. The default margin is zero. (Note that this value is the default margin; any SVF control code in the data stream can change this value.)

SWITCH

Specifies whether activation of the six or eight lines-per-inch, single-space or double-space, and mono-case or dual-case switches is reported as status in SMSDST. If you omit this operand, status is not reported.

UDDPC1

Optional operand that identifies the device to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an X (to denote hexadecimal) or the usual single quotation marks used with the X in normal notation. The default value is 00.

DCA5210 -- Define a 5210 Printer

The DCA5210 macro, with a DCAPORT macro, defines a 5210 printer attached through the device cluster adapter.

Name	Operation	Operand
label	DCA5210	[PS={n 0}] [,TM={n 0}] [,SWITCH= {Y N}] [,UDDPC1={value 00}]

label

Is the name (1-8 alphameric characters) for this device description, and is used in a DCAPORT macro.

PS

Is a decimal number that specifies the number of lines printed on a page. The valid range is 0-255. If zero is specified, no page size is used. The default page size is zero. (Note that this is the default page size, any SVF (Set Vertical Format) control code in the data stream can change the page size.)

TM

Is a decimal number that specifies the number of lines to be skipped for a top margin before the first printed line on a page. The valid range is 0-255. If you specify zero, no skip is performed. The default margin is zero. (Note that this value is the default margin; any SVF control code in the data stream can change this value.)

SWITCH

Specifies whether activation of the six or eight lines-per-inch, single-space or double-space, and mono-case or dual-case switches is reported as status in SMSDST. If you omit this operand, status is not reported.

UDDPC1

Optional operand that identifies the device to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an X (to denote hexadecimal) or the usual single quotation marks used with the X in normal notation. The default value is 00.

DCA4704 -- Describe a 4704 Display

The DCA4704 macro instruction describes a 4704-2 Display device to the 4700 controller.

Name	Operation	Operand
label	DCA4704	MODEL={21 22}, OUTRTBL=label-out [,TRTLIST=label-list] [,TRTBHDR=label-trt] [,MSTRTBL=(msintrt,msoutrt)] [,PINTBL=pinintrt] [,EOMSET=(X'xx' X'FF')] [,CURSOR={N Y R}] [,ERTLS={Y N}] [,OPTIONS=(option[,...])] [,FEATURE=MSRE] [,UDDccc={value 00}]

label

A symbolic name (1-8 alphameric characters) used in the DCA4704 macro instruction.

MODEL

Specifies the appropriate model number. Code 21 if you want the 480-character screen model in your configuration or 22 if you want the 1920-character screen model.

OUTRTBL

Specifies the label of the OUTRTBL macro that describes the output translation table to be used with the display screen.

TRTLIST

Specifies the label of the DCA4704 macro that describes the 4704-2 internal translation tables. You use this operand only if the 4704-2 is to be used in local keytracking mode. If omitted, and local keytracking mode is used, the device assumes its default internal translation tables.

TRTBHDR

Specifies the label of the TRTBHDR macro that describes the input translation table header to be used with the keyboard. If you omit this operand, the universal translation table is used.

MSTRTBL

Specifies the labels of the magnetic stripe reader-encoder translation tables, as follows:

msintrt specifies the label of the MSTRTBL macro that defines the input translation table for the magnetic stripe reader.

msoutrt specifies the label of the MSTRTBL macro that defines the output translation table for the magnetic stripe encoder.

PINTBL

Specifies the label of the PINTBL macro that defines the input translation table to be used on data read from the PIN keypad.

EOMSET

Specifies a one-byte hexadecimal number that indicates the value of the EOM set-selection for the keyboard while in controller-tracking mode. X'FF' is the default value.

CURSOR

Specifies how to handle the display cursor:

Y — Cursor is visible at all times

N — Cursor is never visible

R — Cursor is visible only when an application program is reading from the terminal

R is the default value.

ERTLS

Specifies whether:

Y — A read is posted complete without status as soon as the input segment is full.

N — Wrong-length status is indicated when the next character is entered after the input segment is full.

The default value is N.

OPTIONS

Specifies the 4704-2 shift option:.

NSML Specifies that this device is not to be used to log onto the system monitor ("No System Monitor Logon").

NSCC After you press the Cancel key, the keyboard remains in the current shift case.

If you omit this operand, the keyboard reverts to Case 1 after you press the Cancel key.

FEATURE

Specifies the 4704-2 attached feature:

MSRE Indicates that a magnetic stripe encoder is attached.

If you omit this operand, no magnetic stripe encoder is generated.

UDDccc

Optional operand (or specification of multiple operands) that identifies the device component (or components) to which the user-defined data is to apply. The *ccc* represents a two-character or three-character qualifier that generically describes the device, as follows:

UDDKB — Keyboard

UDDDS — Display

UDDMS — Magnetic stripe encoder

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DEFADDR -- Specify the Device Address

The DEFADDR macro instruction associates devices with slot assignments. One DEFADDR macro is necessary for each assigned base slot address. The first DEFADDR statement with a DEVxx04 labeldev in the CPGEN process becomes the “control operator” 4704 or 3604 at start time. If that 4704 or 3604 is not active, the remaining 4704 and 3604 devices are checked in the order that their DEVxxxx DEFADDR statements occur in CPGEN. Normally, the first DEVxx04 DEFADDR statement is for loop 1, address 1. For the 3606 and 3608, as many as 16 components can be associated with a single base slot address.

Name	Operation	Operand
label	DEFADDR	(labeldev[, {sa, ... sa-sa}][, poolname][, ...], ADDR=(looplabel, tla, mod))

label

Is a name (1-8 alphameric characters) for this address group definition. You use this label in the TERM operand of the STATION macro to assign the address group to the station.

labeldev

Is a name (1-8 alphameric characters) of a DEVxxxx macro, and is the definition of the terminal or terminals to be assigned to this address group.

sa

Is decimal number (1-15) that indicates the subaddresses assigned to these devices. These subaddresses correspond to the addresses represented by the subaddress switches in the shared slot devices. You can specify sequential subaddress assignments by using a hyphen (for example, 1, 2, 3 or 1-3). For multiple-component devices, such as the 3608, only the subaddress of the first component is necessary (must be an even number).

A *labeldev* without an associated subaddress is assumed to refer to a device with a fixed hardware address. If the *labeldev* reference is to any nonslot sharing device, then the subaddress specification is invalid, and the controller ignores it. The *labeldev* references to slot-sharable devices must specify the assigned subaddress.

If the DEV3615 macro specifies OPTION=MG1586 (address sharing), an applicable subaddress specification is necessary on the DEFADDR macro. If you do not specify OPTION=MG1586 in the DEV3615 macro, do not specify a 3615 subaddress in the DEFADDR macro.

poolname

A symbolic name (1-8 alphameric characters) that matches the name of the device pool to which this device is assigned. That is, *poolname* matches the pool name specified on the DPOOL parameter of the STATION macro.

Note: Some RPQ devices (not listed in this chapter) are not supported as pool devices. If you define such a device as a pool device, a warning message appears, and the CPGEN continues. However, the device will not be in a pool.

looplabel

Is a name (1-8 alphanumeric characters) of the label on the LOOPS macro.

tla

Is a decimal number (1-16) indicating the terminal address assigned to this terminal group.

mod

Is a decimal number specifying the modulus value of this terminal group. The valid modulus values are 0, and 2-14.

Notes:

1. The *mod* operand cannot be 3 or 5 if you specify the *tla* operand as 1.
2. If you specify *mod* as 0, the loop allocation is one slot per framing sequence. The modulus value and terminal address are used in determining the loop allocation for a terminal.
3. If you are using LCF, refer to “LCF Interfaces and Limitations” in Chapter 4 for LCF restrictions when coding DEFADDR.

DEFSBF -- Define System Image Buffers

The DEFSBF macro instruction allocates a system image buffer to be used to transmit data between the 4704 Display Models 2 and 3 and the controller.

Name	Operation	Operands
------	-----------	----------

[label]	DEFSBF	r1,c1
---------	--------	-------

label

Is a symbolic name (1-8 alphanumeric characters).

r1,c1

Specify the row and column size of the image buffer to be generated.

DEFSPA -- Define Scratch Pad Area

The DEFSPA macro defines and identifies a scratch-pad area for expanded user storage.

Name	Operation	Operand
	DEFSPA	SIZE=n [,ID={id _}]

SIZE

Specifies the size, in bytes, of the scratch-pad area, in the range 2 through 65,534 bytes. If you specify an odd number of bytes, DEFSPA rounds up to the next even number.

ID

Specifies the optional identification of this scratch-pad area, in the range 0-255. The default value is 1. If you define more than one scratch-pad area, each must have a unique ID.

DEV3604 -- Describe a 3604

You use the DEV3604 macro instruction with the DEFADDR macro to describe a 3604 to the controller.

Name	Operation	Operand
label	DEV3604	MODEL={1 2 3 4 5 6 7} [,OUTRTBL=labelout] [,TRTBHDR=labeltrt] [,OPTIONS={{option[,...]}]} [,MSLITE={1 2 3}] [,EOMSET={X'xx' X'FF}] [,PINTBL=labelpin] [,MSTRTBL=(msintrt,msoutrt)] [,CURSOR={Y N R}] [,ERTLS={Y N}] [,PRTY={Y N}] [,UDDccc={value 00}]

label

Is name (1-8 alphameric characters) for this device description. You use this label in the DEFADDR and LDA macro instructions.

MODEL

Is the model number of the 3604.

OUTRTBL

Specifies the output translation table to be used for the display of data. Replace *labelout* with the name of the OUTRTBL macro. If you do not specify *labelout*, the system uses the 175-character translation table.

TRTBHDR

Specifies the input translation table for data entered at the keyboard. Replace *labeltrt* with the name of the TRTBHDR macro. If you do not specify *labeltrt*, the system uses the universal translation table.

OPTIONS

Specifies 3604 options, where *option* can be:

NSML	Specifies that this device is not to be used to log onto the system monitor (“No System Monitor Logon”).
4905	Specifies that a magnetic stripe reader-encoder is attached to this device.
MG1586	Allows a 3604 to share its base address with another terminal.
9641	See RPQ 7B0412 for information regarding the backward write mode.
NSCC	Specifies that after you press the Cancel key the keyboard remains in the current shift case. The default is that the keyboard returns to shift case 1 after you press the Cancel key.

MSLITE

Is an indicator on the 3604 that specifies that the magnetic stripe encoder is ready to use. You should specify MSLITE only if you code `OPTIONS=4905`. If you specify 4905 and omit MSLITE, the default value is 3.

EOMSET

Specifies a one-byte hexadecimal number that indicates the value of the EOM set selection mask for the 3604 keyboard. The default value is X'FF'.

MSTRTBL

Specifies the input and output translation table definition macros to be used by a magnetic stripe reader-encoder attached to this 3604. Set *msintrt* to the label of the MSTRTBL translation table for the magnetic stripe reader. Set *msoutrt* to the label of the MSTRTBL macro that defines the output translation table for the magnetic stripe encoder.

CURSOR

Specifies how the display cursor displays:

Y —	The cursor is to remain on the screen from the first I/O operation until the 3604 is powered off.
N —	The cursor is not used.
R —	The cursor is to remain on the screen only while the application read is in progress.

ERTLS

Specifies how the LREAD is to be posted:

- Y — The LREAD operation will be posted complete without status when the input segment is full.
- N — The LREAD will be posted with a wrong-length status when the next character is entered on the keyboard after the segment is full. The default is N.

PINTBL

Refers to the label of a specific PINTBL configuration macro when a specific translation table is required for this device. If you do not refer to a specific table, that device uses the first PINTBL in the configuration. The PIN Pad is an RPQ on the 3604; see the *PIN Pad Custom Feature Description RPQ 8K0578*, GA27-2810, for more information.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

UDDccc

Optional operand (or specification of multiple operands) that identifies the device component (or components) to which the user-defined data is to apply. The *ccc* represents a two-character or three-character qualifier that generically describes the device, as follows:

- UDDKB — Keyboard
- UDDDS — Display
- UDDMS — Magnetic stripe reader-encoder

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DEV3606 -- Describe a 3606

You use the DEV3606 macro instruction with the DEFADDR macro to describe a 3606 terminal to the controller. You must code this instruction for the correct operation of the 3606 terminal.

LCF does not support the DEV3606 macro.

Name	Operation	Operand
label	DEV3606	TRTBHDR=labeltrt,OUTRTBL=labelout [,MSTRTBL=labelms] [,PRTY={Y N}] [,UDDFST={value 00}]

label

Is a name (1-8 alphameric characters) for this device description. This label is used in the DEFADDR and LDA macro instructions.

TRTBHDR

Is the label of the TRTBHDR macro that specifies the input translation table to be used with the keyboard.

OUTRTBL

Is the label of the OUTRTBL macro that indicates the output translation table to be used with the display screen.

MSTRTBL

Is the label of the MSTRTBL macro that specifies the translation table to be used with the magnetic stripe reader.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

UDDFST

Optional operand that identifies the device component (financial services terminal, in this case) to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an X (to denote hexadecimal) or the usual single quotation marks used with the X in normal notation. The default value is 00.

DEV3608 -- Describe a 3608

You use the DEV3608 macro instruction with the DEFADDR macro to describe a 3608 terminal to the controller. This instruction must be coded for the correct operation of the 3608 terminal.

LCF does not support the DEV3608 macro.

Name	Operation	Operand
label	DEV3608	TRTBHDR=labeltrt,OUTRTBL=(labelout1,labelout2) [,MSTRTBL=labelms] [,AVGML={n 100}] [,FORMWD={n 75}] [,OFFSET=([begin 6] [, {pages 0}])] [,OPTIONS=OCR] [,PRTY={Y N}] [,UDDccc={value 00}]

label

Is a name (1-8 alphanumeric characters) for this device description. This label is used in the DEFADDR and LDA macro instructions.

TRTBHDR

Is the label of the TRTBHDR macro that specifies the input translation table to be used with the keyboard.

OUTRTBL

Specifies the output translation tables for the display screen and printer. Replace *labelout1* with the label of the OUTRTBL macro that defines the output translation table for the display screen. Replace *labelout2* with the label of the OUTRTBL macro that defines the output translation table for the printer.

AVGML

Is a decimal number indicating the average number of characters for each write operation to the printer. This value determines the size of the storage buffer allocated to this device during the generation of control blocks.

FORMWD

Is a decimal number that specifies the maximum number of 10-pitch character columns for each printer form. The default value is 75.

MSTRTBL

Is the name of the MSTRTBL macro that specifies the translation table to be used with the magnetic stripe reader.

OFFSET

Is a decimal number or numbers indicating the format of the printer forms:

- begin** Is the number of 10-pitch columns to be skipped before printing is to begin.
- pages** Is the number of 10-pitch columns to be skipped between each three-line print operation.

OPTIONS

- OCR** Indicates that the top-line print wheel is 7-pitch characters as opposed to 10-pitch.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at a single loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

UDDccc

Optional operand (or specification of multiple operands) that identifies the device component (or components) to which the user-defined data is to apply. The *ccc* represents either FST or PC1 as follows:

- UDDFST — Financial service terminal
UDDPC1 — Printer

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an X (to denote hexadecimal) or the usual single quotation marks used with the X in normal notation. The default value is 00.

DEV3610 -- Describe a 3610

You use the DEV3610 macro instruction with the DEFADDR and LDA macros to describe a 3610 to the controller.

Name	Operation	Operand
label	DEV3610	MODEL={1 2 3 4 5 12 13} ,OUTRTBL=labelout [,PS={n 0}] [,WL={n 0}] [,SHARED={Y N}] [,CONSHR={Y N}] [,FT={CTAUTO CT CN}] [,TIMEOUT={5 n}] [,OPTIONS=MSG1586] [,DEBLSE={n 10}] [,PRTY={Y N}] [,UDDPC1={value 00}]

label

Is a name (1-8 alphanumeric characters) for this device description. This label is used in the DEFADDR and LDA macro instructions.

MODEL

Is the model number of the 3610.

OUTRTBL

Is the label of the OUTRTBL macro that specifies the output translation table to be used with the print wheel of the printer.

PS

Is a decimal number that specifies the number of lines printed on a page. The range of applicable values is 0-127. If you specify 0, no page size is used. The default value is 0.

WL

Is a decimal number that specifies the line number of a line used as a warning line. Applicable values range from 0 through the value of the PS operand.

SHARED

Specifies whether the 3610 can be shared.

CONSHR

Indicates whether concurrent sharing of the 3610 is permissible.

FT

Specifies whether the forms type is continuous (CN), cut (CT), or automatic-start (CTAUTO) in cut-forms mode. The default value is CN.

TIMEOUT

Specifies the number of seconds (5-20) that the printer motors continue to run after the end of a printing operation. The default value is five seconds.

OPTIONS=MG1586

Allows the 3610 to share the terminal address (base address) of an associated device, such as a 3604. You associate the 3610 with a 3604 by referring to both the DEV3610 macro and the DEV3604 macro in a DEFADDR macro. The OPTIONS operand in the DEV3604 macro must also specify MG1586.

DEBLSE

Is a decimal number that specifies the number of lines to be printed before the forms are debloused (release the forms clamp momentarily to release the paper tension). The applicable range of values is 1-255, and 10 is the default value. The frequency of deblousing is a function of the number of copies of the form being printed. Too frequent deblousing can reduce the printer throughput. If you specify 0, no deblousing occurs.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

UDDPC1

Optional operand that identifies the device component (a printer, in this case) to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an X (to denote hexadecimal) or the usual single quotation marks used with the X in normal notation. The default value is 00.

DEV3611 -- Describe a 3611

You use the DEV3611 macro instruction with the DEFADDR and LDA macros to describe a 3611 passbook printer to the controller.

Name	Operation	Operand
label	DEV3611	MODEL={1 2} ,OUTRTBL=labelout [,PS={n 1}] [,CFOLD=([begin 0] [, {skip 2}])] [,OFFSET=([steps 0] [, {lines 0}])] [,SHARED={Y N}] [,OPTIONS=MG1586] [,PTY={Y N}] [,UDDPC1={value 00}]

label

Is a name (1-8 alphameric characters) for this device description. This label is used in the DEFADDR and LDA macro instructions.

MODEL

Is the model number of the 3611.

OUTRTBL

Is the label of the output translation table to be used with the 3611 passbook printer.

PS

Is a decimal number specifying the number of printed lines on a page. The applicable range of values is 1-41.

CFOLD

Is a decimal number or numbers that specify how to handle a horizontal fold in a passbook.

begin — Is the number of the last line to be printed before the center fold.

skip — Is the number of lines to be skipped to bypass the center fold.

The default value for the skip operand is 0. If you specify *begin* but omit *skip*, 2 is the default for the skip operand. If you omit coding the CFOLD parameter entirely, the default value for both *begin* and *skip* is zero.

OFFSET

Is a decimal number or numbers specifying the initial horizontal alignment of the passbook.

steps — Indicates the number of steps (12 steps are in a line) that a passbook must be spaced over on insertion to move it to an integral print-line boundary.

lines — Specifies the number of full lines that a passbook must be moved from the first integral line boundary to the first logical print line (0-31).

SHARED

Specifies whether the 3611 can be shared.

OPTIONS=MG1586

Allows the 3611 to share the terminal address (base address) of an associated device, such as a 3604 keyboard display. The 3611 is associated with a 3604 by referring to both the DEV3611 macro and the DEV3604 macro in a DEFADDR macro. The OPTIONS operand in the DEV3604 macro must also specify MG1586. If the 3611 is sharing an address with a 4704, you need not code this option on the DEV4704 macro.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

UDDPC1

Optional operand that identifies the device component (a printer in this case) to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an X (to denote hexadecimal) or the usual single quotation marks used with the X in normal notation. The default value is 00.

DEV3612 -- Describe a 3612

You use the DEV3612 macro instruction in conjunction with the DEFADDR and LDA macros to describe a 3612 document and passbook printer to the controller.

Name	Operation	Operand
label	DEV3612	MODEL={1 2 3 12 12P 13 13P} ,OUTRTBL=(labelout1,labelout2) [,PS={{[n 0][,n 1]}}] [,WL={n 0}] [,CFOLD=([begin 0] [, {skip 2}])] [,OFFSET=([steps 0] [, {lines 0}])] [,SHARED={Y N}] [,CONSHR={Y N}] [,FT={CTAUTO CT CN}] [,TIMEOUT={5 n}] [,OPTIONS=MG1586] [,DEBLSE={n 10}] [,PRTY={Y N}] [,UDDccc={value 00}]

label

Is a name (1-8 alphameric characters) for this device description. This label is used in the DEFADDR and LDA macro instructions.

MODEL

Is the model number of the 3612. Specify the suffix "p" for a passbook printer.

OUTRTBL

Specifies the output translation tables for the passbook and document printers. Replace *labelout1* with the label of the OUTRTBL macro that defines the output translation table for the document printer. Replace *labelout2* with the label of the OUTRTBL macro that defines the output translation table for the passbook printer.

PS

The first operand is a decimal number that specifies the number of lines printed on a page. The applicable range of values is 0-127. If you specify 0, no page size is used.

The second operand indicates the page size of the passbook printer. The applicable range of values is 0-41.

WL

Is a decimal number that specifies the line number of a page used as a warning line. The valid range of values is 0 through the value of the PS operand.

CFOLD

Is a decimal number or numbers that specify how a horizontal fold in a passbook is to be handled.

begin — Is the number of the last line to be printed before the center fold.

skip — Is the number of lines to be skipped to bypass the center fold.

If you do not specify CFOLD, 0 is the default for the skip operand. If you specify *begin* but omit *skip*, 2 is assumed for the skip operand. If you omit coding the CFOLD parameter entirely, the default value for both *begin* and *skip* is zero.

SHARED

Specifies whether the 3612 can be shared between two operators.

OFFSET

Is a decimal number or numbers specifying the initial horizontal alignment of the passbook.

steps — Indicates the number of steps (12 steps are in a line) that a passbook must be spaced over on insertion to move it to an integral print-line boundary.

lines — Specifies the number of full lines a passbook must be moved from the first integral line boundary to the first logical print line (0-31).

CONSHR

Indicates whether concurrent sharing of the 3612 is permissible.

FT

This operand specifies whether the forms type is continuous (CN), cut (CT), or automatic start (CTAUTO) in cut-forms mode.

TIMEOUT

Specifies the number of seconds (5-20) that the printer motors continue to run after the end of a printing operation. The default value is 5 seconds.

OPTIONS=MG1586

Allows the 3612 to share the terminal address (base address) of an associated device, such as a 3604. You associate the 3612 with the 3604 by referring to both the DEV3612 macro and the DEV3604 macro in a DEFADDR macro. The OPTIONS operand in the DEV3604 macro must also specify MG1586.

DEBLSE

Is a decimal number that specifies the number of lines to be printed before the forms are debloused (release the forms clamp momentarily to release paper tension). The applicable range of values is 1-255, and 10 is the default value. The frequency of deblousing is a function of the number of copies of the form being printed. Too frequent deblousing can reduce the printer throughput. If you specify 0, no deblousing occurs.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with **PRTY=Y** has preference over I/O for a device with **PRTY=N**. **Y** is the default value.

UDDccc

Optional operand (or specification of multiple operands) that identifies the device component (or components) to which the user-defined data is to apply. The *xxx* represents a two-character or three-character qualifier that generically describes the device, as follows:

UDDPC2 — Printer (the second of a two-component printer)

UDDPC1 — Printer (the first of a two-component printer)

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DEV3614 -- Define a 3614

You use the DEV3614 macro to define a 3614 Consumer Transaction Facility.

Name	Operation	Operand
label	DEV3614	[OPTIONS= {9495 9494}] [,UDDCTF={value 00}]

label

A symbolic name (1-8 alphanumeric characters) for this macro, used in the DEFADDR macro to refer to this device description.

9494

This 3614 does not use encryption.

9495

This 3614 uses the data encryption standard (DES) algorithm (P57).

UDDCTF

Optional operand that identifies the automated teller machine (3614 Consumer Transaction Facility) to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. For example, if you have four identical displays for user-defined data, you might code them UDDDS=01, UDDDS=02, UDDDS=03, and UDDDS=04, respectively, to differentiate among them. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DEV3615 -- Define a 3615

You use the DEV3615 macro to define a 3615 Printer.

Name	Operation	Operand
label	DEV3615	MODEL={1 2} ,OUTRTBL=labelout [,PS={n 0}] [,WL={n 0}] [,SHARED={Y N}] [,CONSHR={Y N}] [,FT={CT CTAUTO CN}] [,OPTIONS=MG1586] [,PRTY={Y N}] [,UDDPC1={value 00}]

label

Is a name (1-8 alphameric characters) for this device description. This label is used in the DEFADDR and LDA macro instructions.

MODEL

Is the model number of the 3615.

1 = printer speed is 60 cps

2 = printer speed is 120 cps

OUTRTBL

Is the label of the OUTRTBL macro that defines the output translation table to be used with the 3615 printer.

PS

Is a decimal number (0-127) that indicates the number of lines on a printed page.

WL

Is a decimal number that specifies the line number of a page used as a warning line. The valid range of values is 0 through the value of the PS operand.

SHARED

Specifies whether the 3615 can be shared. The 3615 can be shared in continuous form (CN) or automatic start in cut-forms (CTAUTO) mode. However, sharing between two operators in cut-forms (CT) mode is not possible because the terminal has only one start key.

CONSHR

Indicates whether concurrent sharing of the 3615 is permissible.

FT

Specifies whether the forms type is cut (CT), continuous (CN), or automatic start (CTAUTO) in cut-forms mode.

OPTIONS=MG1586

Allows the 3615 to share the same loop address as another device, such as a 3604, or other 3615s. The 3615 is associated with the 3604 by referring to both the DEV3615 and DEV3604 in a DEFADDR macro. The OPTIONS operand in the DEV3604 must also specify MG1586.

If the DEV3615 macro specifies OPTION=MG1586 (address sharing), an applicable subaddress specification is necessary on the DEFADDR macro. If you do not specify OPTION=MG1586 in the DEV3615 macro, do not specify a 3615 subaddress in the DEFADDR macro.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

UDDPC1

Optional operand that identifies the printer component to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DEV3616 -- Define a 3616

The DEV3616 macro, with the DEFADDR and LDA macros, describes a 3616 passbook and document printer to the controller.

Name	Operation	Operand
label	DEV3616	[OUTBHDR=label] [,PS=([n 0] [,n 1])] [,WL={n 0}] [,CFOLD=([begin 0] [,skip 2])] [,OFFSET=([steps 0] [,lines 0])] [,SHARED={Y N}] [,CONSHR={Y N}] [,KEY=([N Y] [,N Y])] [,NL={AUTO SPECIFY}] [,PBFOLD={HORZ VERT}] [,MARGIN={n 0}] [,LPI={5 6}] [,CPI=([0 1] [,0 1])] [,LINE=([n 47/57] [,n 83/100])] [,DEFTRT=labeltrt] [,DEFCHAR=lblchar] [,PRTY={Y N}] [,UDDccc={value 00}]

label

Is a name (1-8 alphameric characters) for this device description. This label is used in the DEFADDR and LDA macro instructions.

OUTBHDR

Is the label of the OUTBHDR macro that specifies the IBM World Trade table 5a, special-character translation tables, and user-defined character translation tables that can be sent to the described device.

PS

The first operand is a decimal number that specifies the number of lines printed on a page by the journal print station. The permissible range of values is 0-127. If you specify zero, no page size is used.

The second operand specifies the page size used on the document print station for passbook printing. The permissible range of values is 0-60.

WL

The first operand is a decimal number that specifies the line number of a page used as a warning line on the journal print station. The permissible range of values is zero through the value of the journal print station's PS operand.

CFOLD

Is a decimal number that specifies how to handle the passbook fold.

begin Is the number of the last line to be printed before the horizontal center fold or the number of the last column to be printed before the vertical center fold.

skip Is the number of lines to be skipped to bypass the horizontal center fold or the number of columns to be skipped to bypass the vertical center fold. If you do not specify **CFOLD**, the controller assumes zero for the *skip* operand; if you specify *begin* but don't specify *skip*, the controller assumes 2 for the *skip* operand.

If you omit coding the **CFOLD** parameter entirely, the default value for both *begin* and *skip* is zero.

OFFSET

Is the decimal number or numbers specifying the initial horizontal alignment of the document print station.

steps Indicates the number of steps that the teller must space a document over when inserting it to move it to an integral print-line boundary.

lines Specifies the number of full lines that the teller must move a document from the first integral line boundary to the first logical print line (0-60).

SHARED

Specifies whether the 3616 can be shared.

CONSHR

Indicates whether concurrent sharing of the 3616 journal print station is permissible.

KEY

The first operand specifies whether a Start key must be pressed to permit journal print station printing. The second operand specifies whether a Start key is necessary for document print station printing.

N = Autostart

Y = Start key necessary

NL

Indicates whether the 3616 performs an automatic new line function, after printing the last character specified by the **LINE** operand.

AUTO — Automatic new line function

SPECIFY — No automatic new line function. The program must supply the Newline function. If the program fails to perform the Newline function, a line-length-exceeded status indication (X'0101') returns on the **LWRITE** request.

PBFOLD

Specifies whether the passbook has a vertical or horizontal fold.

VERT — Vertical-fold passbook

HORZ — Horizontal-fold passbook

MARGIN

Specifies the decimal number of document print station columns to skip before printing the first character of each line. The margin value cannot exceed the line length specified in the LINE operand, and the MARGIN value plus the LINE value cannot be more than the maximum number of characters allowable per line (83 or 100).

LPI

Specifies the print-line density of the document print station as 5 lines per inch or 6 lines per inch.

CPI

The first operand specifies the character density of the journal print station.

The second operand specifies the character density of the document print station.

0 - 10 Characters per inch
1 - 12 Characters per inch

LINE

The first operand specifies the line length at the journal print station. The range of values is as follows:

1 - 47 if 10 cpi specified
1 - 57 if 12 cpi specified

Do not use LINE=0 to skip lines without printing. LINE=0 defaults to the maximum line length of 47 or 57.

The second operand specifies the line length at the document print station. The range of values is as follows:

0 - 83 if 10 cpi specified
0 - 100 if 12 cpi specified

DEFTRT

Specifies that the World Trade table 5a translation table is to be loaded into the 3616 as a default. If you omit this operand, no special-character table is loaded. *labeltrt* is the label of the OUTSPEC macro defining the table.

DEFCHAR

Specifies the user-defined character table to be loaded into the 3616 as a default. If you omit this operand, no user-defined character table is loaded.

lblchar - is the label of the CHARDEF macro that defines the user character table.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

UDDccc

Optional operand (or specification of multiple operands) that identifies the device component (or components) to which the user-defined data is to apply. The *ccc* represents either PC1 or PC2 as follows:

UDDPC1 — Printer (the first of a two-component printer)

UDDPC2 — Printer (the second of a two-component printer)

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DEV3624 -- Define a 3624

You use the DEV3624 macro to define a 3624 Consumer Transaction Facility.

Name	Operation	Operand
label	DEV3624	[UDDCTF={value 00}]

label

A symbolic name (1-8 alphameric characters) for this macro, used in the DEFADDR macro to refer to this device description.

UDDCTF

Optional operand that identifies the automated teller machine (3624 Consumer Transaction Facility) to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DEV4704 -- Describe a 4704

You use the DEV4704 macro instruction with the DEFADDR macro to describe a 4704 to the controller.

Name	Operation	Operand
label	DEV4704	MODEL={11 12},OUTRTBL=labelout, CHARSET={US KATAKANA} [,TRTBHDR=labeltrt] [,PRTY{Y N}] [,OPTIONS=({4905 JUCC 9641 NCSR NSCC})] [,EOMSET={X'xx' X'FF'}] [,PINTBL=labelpin] [,MSTRTBL=(msintrt,msoutrt)] [,ALITE={SYS 1 2 3}] [,CURSOR={Y N R}] [,ERTLS={Y N}] [,UDDccc={value 00}]

label

Is a required symbolic name (1-8 alphanumeric characters) for this device description. You use this label in the DEFADDR and LDA macro instructions.

MODEL

Required operand that specifies the model number of the 4704 display. Specify 11 either for a 5-inch screen or if you want 480 characters displayed on a 9-inch screen. Specify 12 to display the full 1920 characters on a 9-inch screen. Do not use 12 if you are defining a 4704 with a 5-inch screen.

OUTRTBL

Required operand that specifies the output translation table to be used when displaying data. Replace *labelout* with the name of the OUTRTBL macro. This operand is necessary.

CHARSET

Required operand that specifies the output translation table character set. This parameter should be compatible with the CHARSET operand of the OUTRTBL macro.

TRTBHDR

Optional operand that specifies the input translation table to be used for keyboard-entered data. Replace *labeltrt* with the name of the TRTBHDR macro. If you do not specify *labeltrt*, the system uses the Universal Translation Table.

PRTY

Optional operand that specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

OPTIONS

Specifies 4704 options, which are:

- 4905 Specifies that a magnetic stripe reader-encoder is attached to this device.
- JUCC Specifies that an additional buffer be allocated for JUCC magnetic-stripe input.
- 9641 See RPQ 7B0412 for information regarding the backward write mode.
- NCSR Specifies "No Clear Screen on Reset". The screen does not clear after a device-reset command.
- NSSC Specifies that after you press the Cancel key, the keyboard remains in the current shift case. The default is that the keyboard returns to shift case 1 after you press the Cancel key.

Address sharing is standard on the 4704.

EOMSET

Specifies a one-byte hexadecimal number indicating the value of the EOM set selection mask for the 4704 keyboard. If you omit this operand, the system uses the value X'FF'.

PINTBL

This refers to the label of a PINTBL macro if a specific translation table is necessary for the PIN keypad attached to this device. If you do not code this operand, this PIN keypad is translated using the first PINTBL defined in the configuration.

MSTRTBL

Specifies the input and output translation table definition macros to be used by a magnetic stripe reader-encoder attached to this 4704. Set *msintrt* to the label of the MSTRTBL macro that defines the input translation table for the magnetic stripe reader. If you do not specify this table, a standard translation table is used. Set *msoutrt* to the label of the MSTRTBL macro that defines the output translation table for the magnetic stripe encoder.

ALITE

Specifies which indicators, when on, will sound the alarm. SYS corresponds to the Check light on a 3604.

CURSOR

Specifies how the display cursor displays:

- Y — The cursor is to remain on the screen from the first I/O operation until the 4704 is powered off.
- N — The cursor is not used.
- R — The cursor is to remain on the screen only while the application read is in progress. This is the default value.

ERTLS

Specifies when the LREAD operation is posted complete:

- Y — LREAD is posted complete without status when the input segment is full.
- N — READ is posted with a wrong-length status when the next character is entered after the segment is full. This is the default value.

UDDccc

Optional operand (or specification of multiple operands) that identifies the device component (or components) to which the user-defined data is to apply. The *ccc* represents a two-character or three-character qualifier that generically describes the device, as follows:

- UDDKB — Keyboard
- UDDDS — Display
- UDDMS — Magnetic stripe reader-encoder

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

Configuration Notes: DEV4704 has no MSLITE operand. The 4704 magnetic device indicator lights are controlled by the terminal rather than by the program.

If you are converting from a DEV3604 macro, be sure to change the TYP A and SHFT operands on the TRTBHDR macro. The scan codes are not the same.

The following example defines a 4704 (small screen) with a 112-key U.S. keyboard.

```
KDS3  DEV4704  MODEL=11,OPTIONS=(4905,NCSR),CHARSET=US,
      OUTRTBL=OUT47,TRTBHDR=TRT4704,CURSOR=Y,ALITE=(1,3)
OUT47  OUTRTBL  DEVICE=4704,CHARSET=US
TRT4704 TRTBHDR  CANCEL=31,INTRTBL=(CASEA,CASEB),
      SHFT=(30,,3D),TYP A=(1F,0E,40,0C)
CASEA  INTRTBL  KYBD=47US112,CASE=1
CASEB  INTRTBL  KYBD=47US112,CASE=2
```

You cannot designate typematic (TYP A=) and downshift (SHFT=) keys for a 50-key keyboard when it is attached to the secondary port (to form part of the 112-key combined keyboard).

The Alt key does not transmit a scan code; it is used for machine functions, such as local reset, cursor blink, and test.

DEV4710 -- Describe a 4710

You use the DEV4710 macro instruction to identify and describe a 4710 to the 4700 controller.

Name	Operation	Operand
label	DEV4710	[OUTBHDR=labelout] [,PS={n 1}] [,WL={n 0}] [,OFFSET={lines 0}] [,SHARED={Y N}] [,KEY={N Y}] [,NL={SPECIFY AUTO}] [,CPI={1 0}] [,LINE={n 40/48}] [,DEFTRT=labeltrt] [,DEFCHAR=lblchar] [,PRTY={Y N}] [,UDDPC1=value 00]

label

Is the required symbolic name (1-8 alphanumeric characters) for this device description. This same label is used by the DEFADDR and LDA macro instructions.

OUTBHDR=labelout

Replace *labelout* with the symbolic name of an OUTBHDR macro, which specifies the national-use special-character tables (refer to *Controller Programming Library, Volume 4*) and the user-defined character-translation tables that can be sent to the 4710.

PS

Specifies the default number of printed lines on a page. Code a value in the range 1-4. If you omit this operand, the system assumes a value of 1. The sum of this operand value and the value of the OFFSET operand must not exceed four lines.

WL

Specifies the default line number to be the warning line. Code a value in the range 0 to the value of the PS operand. The default value is 0.

OFFSET

Specifies the default number of print lines that the document must move from the clamping position to the first print line. The sum of this operand value and the PS operand value must not exceed 4 lines. The default value is 0.

SHARED

Specifies whether this printer is shared between two work stations. Code Y if the printer is shared; code N if the printer is not shared. The default value is N.

KEY

Specifies whether a start key is necessary to start printing. Code Y if the printer requires a start key and N if it does not. When a start key is necessary, the operator must press either the Print 1 Key or Print 2 Key to begin printing. The default value is Y.

Note: If you specify SHARED=Y, the start key is necessary, which means that the specification KEY=N is invalid.

NL

Specifies whether the printer skips to the beginning of a new line after printing the last character specified by the LINE operand. Code AUTO to perform the automatic new-line function. Or, code SPECIFY to have the application program supply the new-line function. If you code SPECIFY, but the application program does not provide the new-line function, any attempt to print past the last position of the line results in status indicating that the line length has been exceeded. The default value is AUTO.

CPI

Specifies the default print density. You should code 0 for 10 characters per inch, and 1 for 12 characters per inch. If you omit this operand, the system assumes a value of 0.

LINE

Specifies the default length of the print line:

- 0 to 40 for 10 characters per inch
- 0 to 48 for 12 characters per inch

If you omit this operand or code a 0, the system uses the largest valid length.

DEFTRT

Specifies that national-use special characters (refer to *Controller Programming Library, Volume 4*) are to be used for the default translation table. Replace *labeltrt* with the symbolic name of an OUTSPEC macro that defines the translation table. If you omit this operand, the national use special character table is not sent to the 4710 when it powers on.

DEFCHAR

Specifies the user-defined character table to be used as the default translation table. Replace *lblchar* with the symbolic name of a CHARDEF macro that defines the table. If you omit this operand, the user-defined character table is not sent to the 4710 when it powers on.

PRTY

Specifies the priority of devices sharing a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. Y is the default value.

UDDPC1

Optional operand that identifies the printer component to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DEV4720 -- Describe a 4720

Use the DEV4720 macro instruction to identify and describe a 4720 printer to the 4700 controller.

Name	Operation	Operand
label	DEV4720	MODEL={1 2 3 4} [,PS={n 1}] [,WL={n 0}] [,OUTBHDR=labelout] [,SHARED={Y N}] [,OFFSET={ [steps 0] [,lines 0] }] [,KEY={N Y}] [,NL={SPECIFY AUTO}] [,LPI={6 5}] [,CPI={0 1 2}] [,LINE={n 82/99}] [,FONT={Q R}] [,ADVANCE={Y N}] [,PRTY={N Y}] [,DEFTRT=labeltrt] [,DEFCHAR=lblchar] [,SKEW={3 2 1 0}] [,DEFQUAL=labelqchar] [,UDDPC1=value 00]

label

Required label; it is a symbolic name (1-8 alphameric characters) for this device description. This label is used in the DEFADDR macro instruction and must be specified with an even-number subaddress.

MODEL

Required parameter that specifies the model number of the 4720. Valid specifications are in the range 1-4, where:

Model 1 has cut-forms capability only.

Model 2 has cut-forms and continuous-forms capability.

Model 3 has cut-forms and passbook capability.

Model 4 has cut-forms, passbook, and continuous-forms capability.

OUTBHDR

Optional operand; *labelout* specifies the name (1-8 alphameric characters) of the OUTBHDR macro, which specifies the translation tables and user-defined character-translation tables that can be sent to the device.

PS

Optional operand; a decimal number in the range 0-*n* that specifies the number of printed lines on a page. The value of *n* is 70 at five lines-per-inch and 84 at six lines-per-inch. The sum of this parameter's value and the OFFSET parameter's value must not exceed *n*.

WL

Optional operand; a decimal number that specifies the line number of a page used as the warning line. Valid values are in the range from 0 through the value of the PS operand.

OFFSET

Optional operand that specifies the number of incremental steps and full lines that the document must be moved from the clamping position to the first logical print line as a decimal number. The sum of this parameter's value and the value of the PS parameter must not exceed the maximum page size for the specified LPI (lines-per-inch parameter). At five lines-per-inch, 18 step increments are on each line. At six lines-per-inch, 15 step increments are on each line.

SHARED

Optional operand that specifies whether the 4720 can be shared between two work stations. The default value is N.

KEY

Optional operand that specifies whether a start key is necessary to initiate document printing. Valid specifications are N (automatic starting eliminates the need for a start key) or Y (a start key is necessary).

NL

Optional operand that indicates whether the 4720 performs an automatic new-line function after printing the last character specified by the LINE operand. Valid specifications are:

AUTO Automatic new-line function

SPECIFY No automatic new-line function. The application program must supply a new-line function. If the application fails to do so, status code X'0101' (line length exceeded) returns from the write request.

LPI

Optional operand that specifies the print-line density, as follows:

5 — five lines-per-inch

6 — six lines-per-inch

CPI

Optional operand that specifies the print-character density, as follows:

0 — 10 characters-per-inch

1 — 12 characters-per-inch

2 — 16.6 characters-per-inch

LINE

Optional operand that specifies the line length, as follows:

0-82 if you specify 10 cpi

0-99 if you specify 12 cpi

FONT

Optional operand that specifies the print font, as follows:

Q — Quality print font (do not specify this for a printer that is to print Katakana characters)

R — Regular print font

ADVANCE

Optional operand that specifies whether the continuous forms are to be advanced when a line is advanced on a cut form, as follows:

- Y — Advance the continuous form when advancing the cut form.
- N — Advance the cut form without continuous form advance.

SKEW

Optional operand that specifies the acceptable amount of skew when registering a cut form, as follows:

- 0 — Acceptable skew = 0.75 mm per 100 mm of line length
- 1 — Acceptable skew = 1.50 mm per 100 mm of line length
- 2 — Acceptable skew = 2.25 mm per 100 mm of line length
- 3 — Acceptable skew = 3.0 mm per 100 mm of line length

DEFTRT

Optional operand that specifies the translation table to be loaded into the 4720 as a default. *labeltrt* is the label of the OUTSPEC macro that defines the translation table.

DEFCHAR

Optional operand that specifies the user-defined character table to be loaded into the 4720 as a default. *lblchar* is the label of the CHARDEF macro that defines the user's character table.

PRTY

Optional operand that specifies the priority of devices that share a loop address. When I/O is directed to more than one device at the same loop address, the controller puts these requests in serial order. I/O for a device with PRTY=Y has preference over I/O for a device with PRTY=N. The default value is Y.

DEFQUAL

Optional operand that specifies the user-defined quality character table to be loaded into the 4720 as a default. *labelqchar* is the label of the CHARDEF macro that defines the quality character table.

UDDPC1

Optional operand that identifies the printer component to which the user-defined data is to apply.

The *value* is a two-place hexadecimal figure you specify in the range 00-FF to identify the particular device in question. Note that you do not code an *X* (to denote hexadecimal) or the usual single quotation marks used with the *X* in normal notation. The default value is 00.

DKATBL -- Specify a Dead-Key Table for Accented Characters

The DKATBL macro defines composite characters that are to result from the combination of an accent mark and a letter in the same character space. This usage is primarily for national languages that make use of such notation as the grave and acute accent marks and the circumflex in French, for example, or the tilde and cedilla in Spanish. All values coded in the macro are hexadecimal representations of the characters as they appear in the user's application program, but without the hexadecimal syntactic notation (the X' ... ' is omitted).

These composite characters are distinct entities, each with its own hexadecimal representation (the composites do not result from any sort of combination of the hexadecimal values for individual characters). The values used might well vary from country-to-country, and most certainly vary according to the specific letter and accent notation used.

You code the macro with one section for each accent character. Each such accent is followed by one or more letter-and-composite pairs.

Name	Operation	Operand
label	DKATBL	(accent,letter-and-composite,...) [(,....)]

label

A required symbolic name that corresponds to the label specified in the DKATBL operand of the TRTBHDR macro that uses this table. More than one device of the same type can use the same input translation table header, and more than one translation table header, for various devices, can refer to the same accent table.

accent

Is the hexadecimal value of the accent mark that goes with the following pair of hexadecimal representations, the first for the input letter, and the second for the output composite character. This *accent* value is the same as specified by the INTRTBL macro in relation to the input scan code for the accent key. The INTRTBL macro must define the scan code as a dead key accent by using the DKA attribute keyword. You can code any number of accents, and each can have any number of letter-and-composite pairs.

letter

The hexadecimal value of the letter that is to receive the accent. This value is the same as specified by the INTRTBL macro in relation to the scan code for the input key.

composite

The hexadecimal value of the composite character, that is, the letter and its accent mark considered together as a discrete character. This value is the same as specified by the OUTRBL macro in relation to the output position code for the desired graphic.

Examples

This example supports the grave accent mark and the cedilla, using EBCDIC translation tables and the default dead key accent table.

```

KBDS      DEV4704 MODEL=11,CHARSET=EB1,OUTRTBL=out47,TRTBHDR=IN47
*
OUT47     OUTRTBL DEVICE=4704,CHARSET=EB
*
IN47      TRTBHDR CANCEL=31,INTRTBL=(CASEA,CASEB),SHFT=(30,3D)
*
CASEA     INTRTBL (00,X'79',DKA),          GRAVE IS 'DKA'
           KYBD=4704EB,CASE=1
*
CASEB     INTRTBL (12,X'E0',DKA),          CEDILLA IS 'DKA'
           KYBD=4704EB,CASE=2
*
*

```

A user-specified table can be substituted for the default table by using the DKATBL macro and referring to it from the TRTBHDR macro.

This example shows the use of a grave accent mark with the letters *a*, *A*, and *e*, and the cedilla accent mark with the letter *c*, plus the space character.

The example uses the US translation tables. The US output table does not define accented characters, and therefore, overrides from the OUTRTBL macro are necessary. Although the following table is not absolutely necessary, it is a useful way of setting out all the values involved before coding the macros.

character						--grave--			cedilla		
	a	A	e	c	gra	ced	a	A	e	c	space
input scan code	21	21	13	34	00	12					40
EBCDIC hex	81	C1	85	83	79	E0	44	64	D0	48	40
output pos. hex	61	41	65	63	1A	EE	C0	D0	C1	E8	01
output pos. dec	97	65	101	99	26	238	192	208	193	232	1

The cedilla is typed as an uppercase 6 for this example.

```

KDS3      DEV4704 MODEL=11,CHARSET=US,OUTRTBL=OUT47,TRTBHDR=IN47
*
OUT47     OUTRTBL (97,X'81'),(65,X'C1'),          a, A
           (101,X'85'),(99,X'83'),              e, C
           (1,X'40'),                            space
           (26,X'79'),(238,X'E0'),              grave & cedilla
           (192,X'44'),(208,X'64'),              a, A accented
           (193,X'D0'),                          e accented
           (232,X'48'),                          c cedilla
           DEVICE=4704,CHARSET=US
*
CASEA     INTRTBL (00,X'79',DKA),KYBD=47US112,CASE=1  grave accent
*
CASEB     INTRTBL (12,X'E0',DKA),KYBD=47US112,CASE=2  cedilla accent
*
*
DKA47     DKATBL (79,8144,C164,85D0,4079),          grave- a, A, e, space
           (E0,8348,40E0)                          cedilla- c, space

```

ENDGEN -- End Configuration Program Generation

ENDGEN indicates the end of a controller configuration procedure and allows segments to be initialized with data.

ENDGEN is normally the last configuration macro instruction in every controller configuration procedure and can appear only once.

Name	Operation	Operand
[label]	ENDGEN	[[id],seg,[disp],data),...]

id

Is a decimal number in the range 2-31 that indicates the ID of the logical work station to be initialized. You do not use the *id* operand when you specify global segments 13 or 15; you can omit *id* or code it as null. For a non-global segment 13, the *id* specifies the storage class of the SEGSTOR macro that defines the segment.

seg

Is the number of the segment to be initialized (0A, 0B, 2-13, 15). 0A and 0B indicate initialization of the segment 0 associated with operator A and operator B, respectively. If the station specified is not shared, only 0A is coded for segment 0.

disp

Is a decimal number indicating the location of the data within the segment to be initialized. If you do not specify it, displacement is either 0 or the next sequential location in the segment. A displacement of 0 into segment 0A or 0B corresponds to the high-order byte of register 0. A displacement of 0 into segment 15 is the start of the user-defined storage; the system storage of segment 15 must not be initialized.

data

Is the data that can be specified in the following form:

`ddd'txxx...x'`

`tLnn'txxx...x'`

`tl.nn'txxx...x'`, where:

ddd is a decimal number indicating the number of times the constant is to be generated. If only a single constant is required, this number is not necessary.

t is the notation describing the data between the single quotation marks. This data can be any type of data supported by a standard assembler, and can be denoted, for example, as X for hexadecimal, C for character, or F for fullword.

nn is a decimal number indicating the actual length of the constant

xxx...x is the data that makes up the constant, enclosed in single quotation marks.

The information within the parentheses must not exceed 255 characters, including the type notation, single quotation marks, and commas.

If you need to initialize more segments than you can specify on `ENDGEN`, use the `SEGINIT` macro to initialize the additional segments.

EXTCTR -- Specify Extended Statistical Counters

EXTCTR defines an extended statistical counter and specifies which set of devices or loop is to be assigned to each extended counter. An extended statistical counter maintains a count of the total number of data bytes transmitted and data bytes in error for input devices.

One EXTCTR is necessary for each extended statistical counter desired. You assign each extended counter a logical identification (ID=X'nnnn'), which can relate to an individual device, a group of devices, or a loop. This name enters the ESPEID field of the parameter list created by the DEFESP operand of the COPY instruction, which is referred to by the STATS instruction when it reads a counter.

The maximum number of EXTCTR macros that you can specify is 256.

Extended statistical counters that are defined for a loop to record loop noise errors cannot have additional loop devices assigned to the same counter.

Note: You cannot code the EXTCTR macro when using the local configuration facility.

Name	Operation	Operand
[label] EXTCTR	(loop[,tla,comp]	[,{sa,... sa-sa}])... ,ID=X'nnnn'

loop

Is a decimal number (1-4) indicating the loop number.

If you specify only one loop, or specify the loop with a *tla* value of 0 and a component value of 0, the counter records loop noise errors. Counters specified this way must not have additional loop devices assigned to the same counter. When defining a loop counter, specify only the loop number.

tla

Is a decimal number (1-16) indicating the terminal address assigned to this terminal or terminal group. This address is the first slot assigned to the terminal or terminal group in the 16-slot transmission frame.

comp

Is the component code for the input device or terminal group, as follows:

1 — 4704 or 3604 keyboard

6 — 3606 or 3608 keyboard

8 — 3614 or 3624 consumer transaction facility

sa

Is a decimal number (1-15) indicating the subaddress assigned to a specified terminal when the *tla* is for a terminal group. You can specify either individual subaddresses or a range of subaddresses. If you omit a subaddress or subaddress range, the whole terminal group is assigned to the extended counter.

ID

Is a two-byte hexadecimal (X'nnnn') value that serves as an identifier for the extended statistical counters. This value puts the extended counters table in ascending sequence and identifies the counter to be read out or reset by the STATS symbolic instructions. The values X'0000' and X'FFFF' are reserved and should not be used.

FILES -- Specify Diskette or Disk Files

FILES identifies such things as the number and type of files on the diskette or disk, the indexing scheme used, and the amount of space available. One FILES macro is necessary for each drive. The space available on the primary diskette is occupied by the controller application programs, the permanent file, the temporary files, and user data sets.

FILES is not referred to by, nor does it refer to, any other macro instruction.

Name	Operation	Operand
[label]	FILES	DKT={1 2} DSK={1 2 A B} [,PF={blks 0}] [,TF=([num 0] [,trks 1])] [,INDX1=({1,...n 0})] [,INDX2=({1,...n 0})] [,INDX3=({1,...n 0})] [,INDX4=({1,...n 0})] [,COMF=({1,...n 0})] [,INDXC=({1,...n 0})] [,BUF={Y N}] [,CPYAP= {Y N}] [,EDAM=([{N datasets Y}] [, {buffers 2}] [, {NOALLOC ALLOC}] [, {XRCD NOXRCD}] [, {KEYED NOKEYED}] [, {MB NOMB}] [, {XLDKT NOXLDKT}])] [,LOGTM={Y N}] [,ALTLED={Y N}]

DKT

Specifies whether this FILES macro describes the primary (1) or secondary (2) diskette drive.

You must not code this operand together with the DSK operand for any one FILES macro.

DSK

Optional operand that specifies which disk drive this FILES macro describes. To describe the first disk drive, you can specify either A or 1 (A is preferable). To describe the second disk drive, you can specify either B or 2 (B is preferable).

You must not code this operand together with the DKT operand for any one FILES macro.

If you specify this operand, optional module MB1 is included in the operating diskette image.

Coding B or 2 specifies that this use of the FILES macro refers to the second drive of the Disk Expansion Unit. If you code B or 2, the configuration must also include another FILES macro on which you code either DSK=A or DSK=1.

PF

Is a decimal number specifying the number of sectors allocated to the permanent file on the operating diskette. Fifteen or twenty-six 256-byte blocks are available on each track, depending on the type of diskette. The default value is 0.

TF

Is a pair of decimal numbers that indicate the number of temporary files (*num*) and the total number of temporary file units (*trks*) allocated to these files. You name temporary files sequentially: if you specify two temporary files, they will be TF1 and TF2, while three files will have the names TF1, TF2, and TF3, and so forth, to a maximum of four temporary files. The default values are 0 for the number of temporary files, and 1 for the number of temporary file units. If the FILES macro with DKT=1 specifies more space than is available on the primary drive, only the space available is allocated for temporary files.

If you code the TF operand together with DKT=2 or the DSK operand, you must specify EDAM. Optional module M1E is included in the operating diskette image.

INDX1,INDX2,INDX3,INDX4

Are decimal numbers that specify the IDs of subfiles to be indexed in each of the temporary files. You can specify as many as 60 subfiles for each temporary file. The specified subfiles can be in any order and need not be sequential through the specified range. You can specify sequential subfile IDs by coding the lower subfile ID, a hyphen, and the higher subfile ID (for example, INDX1=1-10). The default value for each unspecified INDX operand is 0 (no subfiles).

COMF

Is a decimal number or numbers (1-4) specifying the IDs of the temporary files that make up the composite file. The default value is 0 (no composite file). COMF is necessary if you specify INDXC.

INDXC

Is a decimal number or numbers specifying the IDs of the subfiles to be indexed in the composite file. You can specify as many as 60 subfiles in any order, and they need not be sequential through the range. To specify a group of sequential subfile IDs, code the lower ID number, a hyphen, and the higher ID number (for example, INDXC=1-10). The default value is 0 (no subfile). The COMF operand is required if INDXC is specified.

BUF

Indicates whether you want to have a read index buffer. This 256-byte buffer space provides a space for reading in the file index record from the temporary file to speed up access of file information. The default value is N.

CPYAP

Optional parameter that specifies whether you want to copy all application programs to disk for use by LLOAD and APCALL functions. Code Y to indicate that you want copies. N (no) is the default.

Note: This parameter is valid only for the first DSK FILES macro. If you code Y, you must also code the EDAM parameter on the DKT=1 FILES macro and on the first DSK FILES macro. If you also code the MB parameter for the first DSK, you can get a substantial performance improvement for the LLOAD and APCALL functions.

EDAM

Specifies whether the Extended Disk and Diskette Access Method (EDAM) is to be included in the system.

EDAM=N

Is the default, and indicates that no EDAM support is necessary for the drive being described. This specification (or default) is not valid if you specify the TF operand together with the DKT=2 or DSK operand.

EDAM=Y

Indicates that EDAM support is necessary for the drive being described. You should code this operand if you include the *datasets* and *buffers* specifications on another FILES macro, and no further requirements exist for the drive specified by this FILES macro. You must specify EDAM with DKT=1 if your configuration creates diskettes, or if you use the host transmission facility either to create or update an operating diskette.

When you include EDAM with the DKT operand, optional modules M1B and MB2 are included in the operating diskette image. When you include EDAM with the DSK operand, optional modules M1B, MB3, M1D, and MB6 are included in the operating diskette image.

If your system creates or changes data sets, or if it creates or changes the operating diskette by using the host transmission facility, you must also specify ALLOC.

datasets

Specifies the number of EDAM data sets that can be opened concurrently by all stations. The *datasets* specifications for all of the FILES macros are added together to derive the total number of data sets that can be open concurrently by all stations for all drives. If you omit this operand, the number of data sets defaults to 2.

If you open a keyed data set, you also cause the implicit opening of any unkeyed data sets that are associated with it. Similarly, the opening of an unkeyed, direct data set that has one or more associated keyed data sets can cause the keyed data sets and any associated unkeyed data sets to be temporarily opened implicitly for necessary updates also. Thus, your determination of how many data sets that can be open concurrently should include any of these associated data sets that might also be opened implicitly.

buffers

Specifies the number of 256-byte buffers to be used by EDAM for the reading and writing of blocks of data from data sets. All *buffers* specifications from all of the FILES macros you specify are added together to define the total number of buffers that are to be available to EDAM. If you omit this operand on all of the FILES macros you code, the number of buffers defaults to be equal to the total number of data sets.

Regardless of the number you specify for this operand, the FILES macro automatically provides for a minimum of two buffers for each diskette drive and four buffers for each disk drive.

NOALLOC

Is the default. It specifies that EDAM's allocate-deallocate support for diskettes is not necessary for this configuration. (No special support is necessary for the allocate-deallocate function of disk drives).

ALLOC

Specifies that EDAM's allocate-deallocate support for diskettes is necessary for this configuration. If you specify this support for either diskette drive, it becomes available for both drives. (No special support is necessary for the allocate-deallocate function of disk drives). When you specify ALLOC, optional modules M1D and MB4 are included in the operating diskette image.

NOXRCD

Is the default. It specifies that the extended record format (ASDS data sets) is not necessary for this drive.

XRCD

Specifies that the extended record format (ASDS data sets) is necessary for this drive. If you specify XRCD for any drive, it becomes available for all drives. Also, optional module MB8 is included in the operating diskette image.

NOKEYED

Is the default. It specifies that keyed access is not necessary for this drive.

KEYED

Specifies that keyed access is necessary for this drive. If you specify KEYED for any drive, it becomes available for all drives. Also, optional modules MB5 and MB6 are included in the operating diskette image.

NOMB

Is the default. It specifies that multiple-block processing is not necessary for this drive.

MB

Specifies that multiple-block processing is necessary for this drive. If you specify MB for either diskette, it becomes available for both diskettes, and optional module M06 is included in the operating diskette image. Similarly, if you specify MB for either disk drive, it becomes available for both disk drives, and optional module MB7 is included in the operating diskette image.

Coding the MB parameter on the first DSK FILES macro results in a substantial performance improvement in the LLOAD and APCALL functions if the system monitor and user application programs reside on disk.

XLDKT

Specifies that you want to use the LDKT reset output pointer instruction of 4700 assembler language. Optional module MBB is also included in the operating diskette image.

NOXLDKT

Specifies that you do not want to use the LDKT reset output pointer instruction of 4700 assembler language. NOXLDKT is the default.

LOGTM

Specifies whether the user messages that are to be written to the system log are to be marked with the time before the message identifier. Unless you specify Y, no time information (the default) appears in the log.

ALTLED

Specifies whether you want disk and diskette alternate LED support. Y denotes that you do, and N, the default, denotes that you do not. If you select this option, the optional module MBC is loaded and the 4700 controller's display light-emitting diodes (LEDs) indicate the operational state of the attached disks and diskettes. This LED illumination alternates between a display for disks and diskettes, and a display for the system, one-second duration for each. If no such display information is available, the LEDs instead display only the system information.

INTRTBL -- Build Input Translation Table

INTRTBL builds or modifies a standard translation table for a keyboard shift case. Except for the reset key, you can specify functions for each key on the keyboard with the translation table. You specify the reset key by using the TRTBHDR macro instruction, which must be assembled before INTRTBL.

Name	Operation	Operand
label	INTRTBL	$\left[\text{(code[,char][,func]} \left[, \left\{ \begin{array}{l} \text{EOF} \\ \text{EOFA} \\ \text{EOFB} \\ \text{EOM} \\ \text{EOMA} \\ \text{EOMB} \end{array} \right\} , \text{mask[,EID]} \right] \right) , \dots \right]$ $[\text{,KYBD= } \left. \begin{array}{l} 3278A \\ 3278G \\ 3278K \\ 36047 \\ 3606 \\ 3608 \\ 4621 \\ 4622 \\ 4623 \\ 4627 \\ 4661 \\ 4662 \\ 4663 \\ 4664 \\ 4662J \\ 4664J \\ 470450 \\ 47US62 \\ 47US77 \\ 47US107 \\ 47US112 \\ 47KAT62 \\ 47KAT77 \\ 47KAT112 \end{array} \right\} [\text{,CASE}=\{0 1 2 3 4\}]]$ $[\text{,SCRANGE}=(\text{low},\text{high})] [\text{,IND}=\text{Y N}]$

label

The label of this macro is used in the TRTBHDR instruction to identify the translation table for this shift case.

code

Is the scan code of the key to be modified. You can associate more than one scan code with the same character string or function.

Notes:

1. The scan codes are hexadecimal digits, but you code them without any descriptors such as X or single quotation marks.

2. Some keyboards can have a motor bar that takes the place of three keys, but is treated as one. The scan code for the motor bar is taken from the middle key of the three keys that the motor bar replaces. See the shaded areas on the keyboards in *Controller Programming Library, Volume 4* for the positions and scan codes of the motor bar on the keyboards. For example, if a 30-key keyboard has the motor-bar feature, the scan code for the motor bar is 1B.
3. The maximum number of scan codes that you can modify in a single configuration is 1024.

You can define scan codes to have any combination of one translation value (char), one function, and one EOM or EOF designation.

If you specify *scancode* but do not specify any of the following optional operands, the scan code defaults to a no-operation.

char

Is a one-byte to seven-byte character string that you want the key to generate. You can specify the code by using either the characters themselves (C'cc') or the EBCDIC code for the characters (X'nnnn'). You must reduce the maximum character string length (7) by 1 if you also specify a function, by another 1 if you also specify EOM or EOF, and by another 1 if you specify 'A' or 'B' on EOF-EOM.

func

Specifies a function for the key to perform. Use the following mnemonic operands to specify the functions:

Operand	Function
CAS1	Shift to case 1
CAS2	Shift to case 2
CAS3	Shift to case 3
CAS4	Shift to case 4
BKSP	Backspace
BKSP2	Two-column backspace
ADVAN	Advance
ADVAN2	Two-column advance
NOOP	Null key

The following functions are for the 3278 and 3279 display devices only:

Operand	Function
ALTCUR	Alternate cursor (3278 or 3279 only)
BLNKCUR	Blinking cursor (3278 or 3279 only)
CLICKER	Clicker key (3278 or 3279 only)
DEFHI	Default highlighting
DKA	Dead-key character position, for accented characters
BLINK	Blinking
REVERSE	Reverse video
UNDRSCR	Underscore
DEFCOL	Default color
BLUE	Blue characters
RED	Red characters
PINK	Pink characters
GREEN	Green characters
TURQS	Turquoise characters
YELLOW	Yellow characters
WHITE	White characters
DEFPS	Default program storage
PS1	Program storage 1
PSSLST0	PSSLST EBCDIC table
PSSLST1	PSSLST APL table

Note: If a shift key (CAS1-CAS4) is to be a transient shift key (that is, a shift key that puts the keyboard in the specified case only for the time the key is pressed), you must also specify its scan code in the SHFT operand of TRTBHDR. The functions described are not supported by the 3606 and 3608 terminals.

EOF,EOFA,EOFB

Specifies that this key is an end-of-field indicator. EOF implies that additional fields will follow in the message and can be used to save the loop overhead of ending one operation and beginning another when processing input fields. If no additional input arrives within about three seconds, a time-out occurs, and the outstanding read operation ends, as for an EOM.

EOM,EOMA,EOMB

Specifies that this key is to be an end-of-message indicator. An EOM key cannot be a repeat-action (typematic) key. Two keys — EOMA for operator A's station and EOMB for operator B's station — indicate which operator is using this key as an EOM indicator when the terminal is shared between two operators.

Note: 3606 and 3608 terminals do not require EOM specification. If you specify EOM on any scan code except X'0F', truncation of the input message occurs when the scan code is read, and subsequent data is lost.

mask

Is a one-byte, nonzero, hexadecimal value (X'xx') associated with the EOM. The mask enables checking for valid EOM characters. If any bit in the mask matches a bit in the EOM's set-selection mask, the EOM character is valid. You initially set the set-selection mask by means of the EOMSET operand in the DEVxxxx macro during the controller configuration procedure. The controller application program can later change the mask by using the DEVPARM instruction.

EID

Specifies that the translation value defined for *char* will be returned to the application program in SMSEID rather than to the input segment that contains the message text. The EOM length (SMSECT) will be zero. Only one byte can return in the SMSEID field; if you specify more, only the first byte returns.

KYBD

Specifies the 4700 feature number or DCA keyboard type.

CASE

Specifies the standard translation table shift case to be used as a base for this translation table. The default value is 1. If you specify 0, no standard translation table base is assumed; instead, the table generated contains all no-operation codes except for the scan codes defined with this INTRTBL macro.

SCRANGE

Is a pair of hexadecimal numbers indicating the desired range of values to be used as scan codes in the translation table for this keyboard. The first value represents the low scan code; the second value represents the high scan code. The creation of a translation table using this operand generates a table of no-operation codes. You put values in the table as you want to, using the *scancode*, *func*, and *char* operands.

You can use the SCRANGE operand, together with the KYBD and CASE operands, to generate an abbreviated or an expanded translation table filled with standard keyboard definitions. When you do this, the controller uses the standard definitions for scan codes that are defined within the range of the standard table. It ignores scan codes outside the standard range, and undefined scan codes become no-operation codes. Be sure that all scan codes match the desired functions.

IND

Specifies whether to display the shift indicator (a little arrow, which represents the keyboard's current shift case) on the screen of a 4704 display device or other keyboard-display devices attached through the device cluster adapter. Y means yes, that the indicator should appear, and N means no, that the indicator should not appear. The default value is N.

LDA -- Specify the Logical Device Address

The LDA macro permits multiple devices that share a single address, or components of multiple-component devices, to be separately assigned to work stations in the configuration. The primary use of this macro is to assign multiple printers that share a single address to multiple work stations.

The LDA macro, which is associated with the STATION, DEFADDR, and DEVxxxx macros, allows device components to be assigned to work stations. Also, multiple devices on shared addresses (slots) can be assigned to or across work stations.

When a single device is shared across stations (by way of the shared terminal A/B station indicator), you should use the normal STATION-DEFADDR-DEVxxxx macro relationships.

Name	Operation	Operand
label	LDA	defaddr,device,[component] ,[{lda,...}] [, {A B}]

label

Is the label used in the TERM operand of the STATION macro.

defaddr

A positional operand that is the label of a DEFADDR macro.

device

A positional operand that is the label of the DEVxxxx macro that is referred to by the DEFADDR macro specified in *defaddr*.

component

A positional operand that specifies the subcomponent of a multiple-component device. The component number is the relative position within the string of components as they appear in the DEFADDR macro. For example:

```
1 = 4704-3604 keyboard  
   3608 keyboard-display-magnetic stripe reader  
   3612 journal printer
```

```
2 = 4704-3604 display  
   3608 printer  
   3612 passbook printer
```

```
3 = 4704-3604 reader-encoder
```

The default for this operand is that all components are assigned to sequential logical device addresses.

lda

A positional operand that specifies the logical device address (0-7) or addresses to be assigned to the specified *component* or device. If you omit this operand, logical device addresses are assigned as follows:

LDA Terminal

0	3614 or 3624 terminal
0	4704 or 3604 keyboard
0	3606 keyboard-display-magnetic stripe reader
0	3608 keyboard-display-magnetic stripe reader
1	4704 or 3604 display
1	3608 printer
2	4704 or 3604 magnetic stripe reader-encoder
3	4710 or 4720 or 3610 or 3612 document printer
3	3616 journal printer
3	3615 administrative terminal printer
4	3611 or 3612 passbook printer
4	3616 document printer
6&7	(reserved)

A or B

If this is a shared terminal, this operand specifies whether A or B is the indicator for the station.

Examples of LDA Use:

The same DEVxxxx label can appear more than once in the DEFADDR macro. When this occurs, the string of components is effectively one continuous series of components. As the following example shows, DEFADDR has a string of four components in a series:

	STATION	ID=2,TERM=(...,JP2,DP2)
	STATION	ID=3,TERM=(...,JP3,DP3)
	.	.
	.	.
	.	.
JP2	LDA	DFATAG,T3616,1,...
DP2	LDA	DFATAG,T3616,2,...
JP3	LDA	DFATAG,T3616,3,...
DP3	LDA	DFATAG,T3616,4,...
DFATAG	DEFADDR	(T3616,2),(T3616,4),ADDRS=(L1,2,4)
T3616	DEV3616	...

If different DEVxxxx labels appear in the DEFADDR macro, the string of components is not regarded as one string but as a group of separate individual strings.

For example, each device referred to below has two components; their component specification is:

JP2	LDA	DFATAG,TERMA,1,...
DP2	LDA	DFATAG,TERMA,2,...
JP3	LDA	DFATAG,TERMB,1,...
DP3	LDA	DFATAG,TERMB,2,...
DFATAG	DEFADDR	(TERMA,2)(TERMB,4),ADDRS=(L1,2,4)
TERMA	DEVxxxx
TERMB	DEVxxxx

LOOPS -- Define Terminal Loops

LOOPS identifies each loop. Loop 1, which has the ID of 1, must have the operands TYPE=L. If you do not specify Loop 1 in a LOOPS macro, it is defined by default as follows:

```
L1 LOOPS ID=1,TYPE=L,BPS=4800
```

For controllers using a BSC3 host link and speeds as high as 4800 bps, the total value of the BPS operands on all LOOPS macros plus the value of the DCL operand of the COMLINK macro must not exceed 16,800.

For controllers using an SDLC host link the total value of the BPS operands on all LOOPS macros must not exceed 16,800.

| The label of the LOOPS macro is referred to in the DEFADDR macro.

Name	Operation	Operand
label	LOOPS	ID=n,TYPE={L R},BPS=nnnn

ID

Specifies the number of the loop (1-4)

TYPE

Specifies whether the loop is local (L) or remote (R). A loop attached through a 3603 adapter is considered remote.

BPS

Is a decimal number that specifies the speed of the loop in bits per second (600, 1200, 2400, or 4800).

MSTRTBL -- Build Magnetic Stripe Translation Table

MSTRTBL builds or modifies a standard translation table for the magnetic stripe encoder or magnetic stripe reader. This function allows the application programmer to define values in the translation table for all scan codes, to flag EOM and EID characters for the 4704 and 3604 devices, or for the 3278 or 3279 devices, and to define characters that are not to be translated as *no-operation* characters.

The 4704 and 3604 devices that support magnetic stripe reading require that X'B' correspond to the Start of Stripe function (SOS) and that S'C' and X'F' correspond to the End of Stripe (EOS) function. The 4704 and 3604 device significance of X'B', X'C', and X'F' cannot be modified and must remain flagged as such in the translation tables. However, the configuration programmer can redefine the *mcode* values to provide translation values (*char*) or EID/EOM/EOF designations (or both), or to remove an EID/EOM/EOF designation. If you do not code the MSTRTBL macro, the controller uses the standard translation table for the magnetic stripe reader-encoder.

The EOM and EID codes for the 3278 and 3279 devices depend on whether the terminals attached through the DCA are processing data streams by using 3270 data stream mapping (DATSM). A non-DATSM 3278 or 3279 device does not recognize X'C' as the EOM-EOS character. On the other hand, 3278 or 3279 devices that do process DATSM data streams recognize X'B' as both the EOM-EID character and the SOS character.

Name	Operation	Operand
label	MSTRTBL	[([mcode] , [char] [, func]) . . . ,] TYPE=({IN OUT} [, NULL])

label

This label is used in the appropriate DEV n and DCA n configuration macros to identify this translation table for the magnetic stripe readers or magnetic stripe encoders attached to those devices.

mcode

Is a binary value that specifies the magnetic stripe code to be modified. You specify this operand as a single digit (0-F) without notation. This operand is necessary if you code TYPE=IN. If you omit it for TYPE=OUT, the *func* is treated as a no-operation code.

char

Is a hexadecimal (X'nn') or character (C'n') data string that assigns the desired value to the *mcode* scancode. This operand is necessary if you code TYPE=OUT. If you omit it for TYPE=IN, the *func* operand is treated as a no-operation code.

func

Assigns a special function to the *mcode* or *char* operands. Specify *func* as one of the following:

EOM — On input, causes the read operation to break, and control returns to the application program; on output, the occurrence of the associated *char* in the data stream signifies the end of output message.

Note: EOM and EID have no meaning for the 3606 and 3608 terminals and is ignored if defined.

EOF — The *mcode* or *char* acts as an end-of-field marker. On a reading operation, the *mcode* or *char* acts as an end-of-field marker. When the macro processor detects the *mcode* or *char*, it causes a three-second delay to occur before the reading operation ends. This delay allows more than one message to be read without ending one read operation and beginning another. If no additional input is read within three seconds, the read operation ends as for an EOM.

NOOP — On input, the *mcode* has no corresponding *char* output in the data stream going to the application program; on output, the *char* is not translated to an *mcode*, effectively deleting the *char*.

EID — Returns the translation value specified for *char* in SMSEID for the application program during input operations, and sets the EOM length in SMSECT to zero. EID also acts as an EOM, ending the read operation and returning control to the application program.

TYPE

Specifies whether the translation table is for magnetic stripe reader input (IN) or for magnetic stripe encoder output (OUT). If you specify NULL, a table of all null values is built with only those entries specified in the *mcode*, *char*, and *func* operands. If you do not specify NULL, the controller uses the default translation tables as a base and modifies them by the specified *mcode*, *char*, and *func* operands.

Programming Notes for 3278 and 3279 Magnetic Stripe Readers

1. EOS character of X'C' causes a red-light error on these devices. To ensure compatibility, use the *mscode*, *char*, and *func* operands to set the EOS character to X'F', as follows:

```
...F,X'7C',EOM),TYPE=IN
```

2. When processing DATSM data streams, the 3278 or 3279 magnetic stripe reader recognizes X'B' as an EOM-EID character and creates a protected, nondisplayed field at the present cursor location in the image buffer. The magnetic stripe reader also notifies the application program to read the stripe data using LREAD NOTRACK and put the data in the protected field. The FCT entries for all fields affected by the program are marked as modified. If the current cursor position is in an attribute location or a protected field when the magnetic stripe reader recognizes the EID, the data does not go into the image buffer.
3. A 3278 or 3279 magnetic stripe reader that tries to read a card having five or fewer characters sets the red error light and rejects the card.
4. Before issuing user-supplied magnetic stripes for use on a 3278 or 3279 magnetic stripe reader, you should verify them with an application program written for that purpose.

| OPTLIST -- Specify Optional Module List

OPTLIST creates an optional module subset list to be used during the IPL procedure instead of entering the options in response to the '00001' prompting message.

LCF does not support the OPTLIST macro.

Name	Operation	Operand
------	-----------	---------

```
[label] OPTLIST module1[,module2,...]
```

module1,module2,... n operand (OPTLIST) n (OPTLIST)

The value you substitute for *module1*, *module2*, (and so on) is a list of the optional modules in the IPL subset list. The *modulen* value is the same as the name parameter on OPTMOD. Use a comma to separate each item after the first item in the list. The maximum number of list items is 50.

OPTMOD -- Specify Optional Module List

OPTMOD enables the application programmer to include the device independent programming facilities into the 4700 subsystem during the configuration procedure.

Name	Operation	Operand
[label]	OPTMOD	{name1 func1} [, {name2 func2}]

name

Is a three-character module name that identifies the desired optional module.

func

Is a symbolic name that associates the desired option to the module name. This symbolic name is translated to the proper optional module identifier.

Valid names and functions are:

Name	Function	
—	COBOL	Loads modules P31, P34, and P68.
M04	----	Color convergence support for 3279 color - APL. You need not specify this operand on the OPTMOD macro when you specify MODEL=8A/8B on the DCA3279 macro.
M06	MBIO	The MBIO optional module supports the multiple-block I/O function for the diskette. You need not specify this operand if you specify MONITOR=EXPBM or CNMBM on the STARTGEN macro or the MB suboperand of the EDAM operand of the FILES macro for the diskette.
M18	----	LU assign function.
M29	----	Dynamic APLOAD extension to APCALL.
M2E	----	SDLC-SNA secondary link-swap function
M36	----	Transient data module
M45	----	FINDAP instruction processor for storage management statistics.
MBB	----	Permits the use of the LDKT reset output pointer of the 4700 assembler language; can also be specified by parameter 7 of the EDAM option on the FILES macro.
MBC	----	Supports the alternate LED function for disk and diskette; can also be specified with the ALTLED parameter of the FILES macro.
P21	LTRT	The P21 algorithm supports the LTRT instruction.
P24	LSEEKP	The LSEEKP optional module supports the LSEEKP and LSEEKPL instructions.

Name	Function	
P25	STATS	The P25 optional module provides the function executor for the STATS symbolic instruction. This operand need not be specified in the OPTMOD macro when an extended statistical counter is defined by the EXTCTR macro.
P26	DECOMP	The DECOMP module supports the DECOMP and DECOMPTB instructions.
P27	COMP	The COMP optional mode supports COMP and COMPTB instructions.
P28	----	Security instruction executors (processors)
P2A	SCRPAD	The SCRPAD optional module supports the scratch- pad (SCRPAD) instruction. You need not specify this operand on the OPTMOD macro when you code the DEFSPA macro.
P2C	INTMR	The INTMR optional module supports the interval timer (INTMR) instruction. You need not specify this operand on the OPTMOD macro when you specify a nonzero value for the INTMR operand on any STATION macro.
P31	----	Zoned decimal operation.
P32	----	LTIMEV instruction executor
P34	----	Provides support for the CRETN and LCONVERT instructions.
P40	----	Diskette FORMAT instruction processor
P41	----	Diskette COMPRESS instruction processor
P42	----	The DPOOL optional module supports the device pool function.
P56	ENCODE- DECODE	Encryption-decryption instruction processor
P57	DES	The DES optional module supports the ENCODE and DECODE instructions using the P57 algorithm. When you specify a DEV3624 macro, P57 need not be included in the OPTMOD macro.
P5C	SORT-MERGE	The SORT-MERGE optional module supports the SORT and MERGE instructions.
P5E	SETDSKT	The SETDSKT optional module supports the SETDSKT instruction.
P68	SUBSET1	The P68 algorithm is required for LEXEC, SCALE, SETX, SETXREG, and TESTX.
P70	DSI	Loads module to support 3270 data stream processing (DATSM).

OUTBHDR -- Define Loadable World Trade Special Character Tables

The OUTBHDR macro specifies which of the World Trade Table 5a: National Use Graphics Character Sets can be loaded into a particular device. You must code a unique OUTBHDR macro for each character set to be used, and code at least one operand.

Name	Operation	Operand
label	OUTBHDR	{[OUTSPEC=(mac1b11,mac1b12...,mac1bln)] [,CHARDEF=(mac1b11[,mac1b12])]}

label

Is a symbolic name (1-8 alphameric characters) used in the OUTBHDR operand of the appropriate device's DEVxxxx macro.

OUTSPEC

Specifies the labels of the OUTSPEC macros, each of which defines a National Use Graphics character set to be used, or — for a 4720 — a loadable translation table.

CHARDEF

Specifies the label (or labels) of a CHARDEF macro (or macros ... a maximum of two) that in turn specifies the loadable user-defined special characters that you want to be used. If you specify two such labels, one is for a CHARDEF macro that defines a regular user-defined special-character table, while the other is for a CHARDEF macro that defines a quality-print user-defined special-character table.

OUTRTBL -- Build Output Translation Table

OUTRTBL builds and modifies the standard output translation table.

Name	Operation	Operand
label	OUTRTBL	$\left[\left(\begin{array}{l} \text{pos}, \text{X}'\text{n}' \end{array} \right), \dots, \right] \text{DEVICE} = \left. \begin{array}{l} 3278 \\ 3604 \\ 3606 \\ 3608 \\ 3610 \\ 3611 \\ 3612 \\ 3615 \\ 4704 \\ 4704M2 \end{array} \right\}$ $, \text{CHARSET} = \text{charset} \left[, \text{DEF} = \text{pos} \right]$

label

Is a required label (1-8 alphanumeric characters) that corresponds to the OUTRTBL operand of a DEVxxxx macro.

pos

Is a decimal number indicating the position number of the character on the print wheel or display to be altered (optional).

X'n'

Is the one-byte EBCDIC value to be associated with the position number specified. More than one of the hexadecimal values specified can correspond to the same position number (optional).

DEVICE

Specifies the model number of the terminal that requires the table. The local configuration facility does not support the Financial Services Terminals (3606 and 3608).

CHARSET

Specifies the character set used by the terminal. The value that you select for *charset* depends on the type and model of the device using the character set. LCF does not support the Financial Services Terminals (3606 and 3608).

DEF

Specifies a default position (*pos*) to be printed when the data contains an EBCDIC character that was not defined in the standard translation table (optional).

Device Type	CHARSET Value	Application Notes
3604	175	All models
	228E	Model 7 only
	228K	Model 7 only
3606	16	Display
3608	16	Display-printer
	45	Display-printer
3610	48	Document printer
	64	Document printer
	96	Document printer
	96K	Model 5 only
3611 and 3612	128	Document printer
	64	Passbook and document printer
	96	Passbook and document printer
3615	128	Passbook and document printer
		Administrative printer
3278	192E	English display
	192K	Katakana display
	192G	Greek display
4704	US	US alphameric and special characters
	KATAKANA	Katakana alphameric and special characters
4704-2	US	US alphameric characters
	KATAKANA	Katakana alphameric characters
	HEBREW	Hebrew alphameric characters
	GREEK	Greek alphameric characters

Figure 3-1. Permissible CHARSET Parameters for the OUTRTBL Macro

OUTSPEC -- Define a World Trade National Use Differences Table Character Set

The OUTSPEC macro specifies one World Trade Table 5a: National Use Graphics Table (as described in the *IBM 3270 Information Display System Character Set Reference*). Each of these character sets describes the special characters unique to each language. You must code one OUTSPEC macro for each language table you want. An OUTSPEC macro can be referred to by one or more OUTBHDR macros.

Two distinct procedures apply to the coding of the OUTSPEC macro, depending on the device to which the macro is to apply:

- Coding OUTSPEC for a 3616 or 4710 printer
- Coding OUTSPEC for a 4720 printer

When coding OUTSPEC for a 3616 or 4710, you must specify the character translation table by using the CHARSET parameter. When coding OUTSPEC for a 4720, however, you have a choice between specifying the translation table by using the CHARSET parameter or by using the CPGID parameter to denote a code-page global identifier instead. The CHARSET and CPGID parameters are mutually exclusive on a 4720 specification. The table that follows the parameter descriptions below gives information for appropriate CHARSET or CPGID specifications.

The syntax is as follows:

Name	Operation	Operand
label	OUTSPEC	<pre>[(pos,x'aa'),...], ,DEVICE={3616 4710 4720} [, {CHARSET=country CPGID=id-number}]</pre>

label

Is a required symbolic name (1-8 alphanumeric characters) that OUTBHDR macros use to refer to the National Use character set specified by the CHARSET or CPGID operand specifications that follow on this macro expression.

pos

This parameter is for the 4720 printer only and applies only to translation tables that you specify by using the CHARSET parameter. It is optional and positional. It must appear first in your macro specification if you choose to code it.

The value for *pos* is a decimal number in the range 1-192 that indicates the position number of the character to be altered in the translation table that you specify. It has a companion parameter, described immediately below, that you must use with it. Using these parameters, or a series of them, overrides the character or characters in each of the positions you denote.

X'aa'

This parameter, the companion to *pos* described above, is a one-byte EBCDIC value that is to replace the character located at the position number *pos* in the translation table.

DEVICE

A required parameter that specifies the device with which this character set is to be associated.

Note: If you specify either the 3616 or 4710 device, you can use only the CHARSET parameter to name the translation table that the device is to use.

If you specify the 4720 printer, you can also choose to use the CPGID parameter instead to specify the translation table. Moreover, you can use the *pos*-and-*X'aa'*-parameter pair to override characters in a translation table specified by the CHARSET parameter.

CHARSET

Specifies by name which translation table is to be loaded into the device ... mutually exclusive with CPGID. You must use this parameter for 3616 or 4710 printers and can also use it for 4720 printers.

CPGID

This parameter is for the 4720 printer only and is mutually exclusive with the CHARSET parameter. It specifies (by code-page global identifier) which translation table is to be loaded into the device. The global identifier is a five-digit number that corresponds to one of the translation tables. However, the Hebrew, Greek, Arabic, and Katakana translation tables have no corresponding CPGIDs.

The following table relates printers, character sets, and code-page global identifiers to each other. An X denotes that the printer indicated in its column heading above supports the translation table for the country listed in its row heading at the left.

CHARSET	CPGID	3616 (no CPGID)	4710 (no CPGID)	4720
US	0037	X	X	X
INT	00256	X	X	X
GERMANY	00273	X	X	X
AUSTRIA	00273	X	X	
BELGIUM	00274	X	X	X
BRAZIL	00275	X	X	X
NORWAY	00277	X	X	X
DENMARK	00277	X	X	
FINLAND	00278	X	X	
SWEDEN	00278	X	X	X
FRANCE	00297	X	X	X
ITALY	00280	X	X	X
JAPAN	00281	X	X	X
PORTUGAL	00282	X	X	X
SPAIN	00283	X	X	X
SPAINSPK		X	X	
UK	00285	X	X	X
CANADA	00276	X	X	X
KATAKANA		X	X	X
YUGOSLAVIA	00321	X	X	X
LATIN	00284			X
TURKEY	00322			X
SOAFRICA	00258			X
HEBREW				X
ARABIC				X
GREEK				X

PINTBL -- Describe a PIN Keypad Translation Table

PINTBL specifies the required translation table for an attached PIN keypad device. The format of the PINTBL macro is:

Name	Operation	Operand
label	PINTBL	$\left[\left(\left(\begin{array}{l} \text{pincode} \\ \text{START} \\ \text{END} \\ \text{FILL} \\ \text{ENCRYPT} \end{array} \right) , \left\{ \begin{array}{l} \text{X'nn'} \\ \text{NOOP} \\ \text{C'c'} \end{array} \right\} [, \text{EOM}] \right) \right] , \dots$

label

Specifies the label by which this macro is named in a DEV4704, DCA4704, or DEV3604 configuration macro.

pincode

Is a one-digit scan code, in the range 0-9. This value represents the scan code received from the PIN pad, the scan code that you want translated to a hexadecimal or character value.

START

Specifies that you want the start-of-PIN code translated to a value. The default value for unencrypted keypads is X'7F'.

END

Specifies that you want to translate the end-of-PIN code. The default value is X'7F'.

ENCRYPT

Specifies that you want to translate the start-of-PIN code for an encrypted PIN pad. The default value is X'7E'.

FILL

Specifies a character to be used as the fill character during tracking. The default value is an asterisk (*).

X'nn'

Is the hexadecimal value of the output, translated value. For example, if you code:

```
PIN1 PINTBL (8,X'08')
```

You assign the value of X'08' to the scan code 8 (08).

C'c'

Assigns a character value to the input code in the same way as the X'nn' operand assigns a hexadecimal code.

NOOP

Indicates that this scan code is not to be put into the translated output stream for the application program.

EOM

Specifies that this scan code will also generate an end-of-message code.

PRIDSP -- Priority Table Definition

PRIDSP specifies the priority tables for use with priority dispatching.

Note: You cannot specify the PRIDSP macro when using the local configuration facility.

Name	Operation	Operand
[label]	PRIDSP	({stid stid-stid} [, ...]) [, ...] [, STARTUP={ON OFF N}] , ID=n

stid

Is any valid station (1-60). You need not specify a station identifier (ID) of 1, which provides access to the system monitor, because system monitor access is automatic.

You specify station IDs in sets of one or more IDs. The IDs can be in any order and can be repeated as often as you want. You can use range notation, — *stid-stid* — to specify ascending, consecutively numbered, station IDs, by specifying the first and last IDs in the range.

All stations in the first set of IDs are tested for dispatch before each station in the second set; all stations in the first two sets are tested before each station in the third set, and so on.

The maximum number of entries in the table is 256.

IDs that you specify in PRIDSP but not in the configuration get deleted from the table.

IDs that you do not specify in PRIDSP but specify instead in the configuration get appended to the table.

STARTUP

Specifies the condition of priority dispatching during startup.

ON — This table's ID is put in GMSPRI, and priority dispatching will be active (GMSPRI status indicator = 1) at startup.

OFF — This table's ID is put in GMSPRI, and priority dispatching will be inactive (GMSPRI status indicator = 0) at startup.

N — This table's ID will not be put in GMSPRI at startup. N is the default.

ID

Specifies the number (1-4) that identifies the table to be used when change priority is ON. This ID is placed in GMSPRI if you specify STARTUP=ON/OFF. ID also represents the table ID referred to by the LCHAP instruction.

Note: When you generate more than one priority table, only one PRIDSP macro can have the table ID in GMSPRI. The other PRIDSP macros must either specify STARTUP=N or not use STARTUP at all.

PSSLST -- Define Programmed Symbol Sets

PSSLST defines programmed symbol sets for devices that support them, and allows two such sets to be grouped together for a device. Programmed symbol sets are valid only for 3278 and 3279 display devices.

Name	Operation	Operand
label	PSSLST	(name1 [, name2])

label

Is a name (1-8 alphameric characters) used by the OUTRTBL or TRTBHDR operand of the appropriate DEVxxxx or DCAxxxx device macro to refer to this PSSLST macro. Every PSSLST macro must have a unique label.

name1[,name2] n operand (PSSLST) n (PSSLST)

These are the names of the OUTRTBL or TRTBHDR macros that define the tables to be exchanged. You must specify at least one name, assumed to be *name1*.

Note: Application programs that invoke the SWAPTT instruction to exchange translation tables cannot operate on devices whose DEVxxxx or DCAxxxx macros refer to a PSSLST macro.

SEGINIT -- Initialize Storage Segments

SEGINIT initializes storage segments with data. SEGINIT must follow ENDGEN in the configuration procedure, and can be used in conjunction with, or in place of, the initialization operand of ENDGEN. You can code SEGINIT as many times as necessary to initialize the storage segments, or you can initialize segments in your application program with the INITSEG instruction, but only for the initial application program.

Name	Operation	Operand
label	SEGINIT	[[id],seg,[disp],data),...]

id

Is a decimal number (in the range 2-60) that indicates the ID of the logical work station to be initialized. You do not use the *id* operand when you specify global segments 13 or 15. You can omit *id* or code it as null. For a non-global segment 13, the *id* specifies the storage class of the SEGSTOR macro that defines the segment.

seg

Is the number of the segment to be initialized (0A, 0B, 2-13, 15). 0A and 0B indicate initialization of the segment 0 associated with operator A and operator B, respectively. If the station specified is not shared, you code only 0A for segment 0.

disp

Is a decimal number indicating the location of the data within the segment to be initialized. If you do not specify it, displacement is either 0 or the next sequential location in the segment. A displacement of 0 into segment 0A or 0B corresponds to the high-order byte of register 0. A displacement of 0 into segment 15 is the start of the user-defined storage; the system storage of segment 15 cannot be initialized.

data

Is the data, which you can specify in the following form:

`ddd'txxx...x'`

`tLnn'txxx...x'`

`tL.nn'txxx...x'`, where:

ddd is a decimal number indicating the number of times the constant is generated. If only a single constant is necessary, this number is necessary.

t is the notation describing the data between the single quotation marks (for example, X for hexadecimal, C for character, F for fullword).

nn is a decimal number indicating the actual length of the constant

xxx...x is the data that makes up the constant, enclosed in single quotation marks

Note: See “LCF Limitations and Interfaces” in Chapter 4 for LCF restrictions when coding the **data** parameter of SEGINIT.

The information within the parentheses must not exceed 255 characters, including the type notation, single quotation marks, and commas.

SEGSTOR -- Specify Segment Storage

SEGSTOR specifies the number and size of storage segments 0 (the optional portion) and 2 through 12. You can also define a non-global segment 13 to be shared by all stations using the same class of storage, or among stations using specified classes of storage. These specifications are associated with a class that is referred to in the STATION controller macro instruction. Alternatively, you can define storage segments in an application program with the DEFSTOR assembler instruction.

Name	Operation	Operand
[label]	SEGSTOR	CLASS=nnn,SEGSIZE=([seg0] ,seg2,...,seg12) [, {S13=size S13ID=class}]

CLASS

Is the decimal number that specifies the class to be assigned to this storage configuration (1-240). This number is used in the SS operand of the STATION macro.

SEGSIZE

Specifies a list of decimal numbers indicating the sequential storage specifications (in bytes) in the segments for each station. The first specification listed does not include the fixed portion of segment 0. If you specify multiple operators for the logical work station using this class storage, the size specified for segment 0 is allocated to each operator.

If you don't need optional storage in segment 0, you can omit the *seg0* operand, but you must precede the segment 2 specification with a comma. On the other hand, you must specify storage for all the other segments (even if the specification is for 0 bytes) in sequence to the last segment you want.

S13

Specifies the size of a shared segment 13, which will be shared by all stations that use the storage class specified by CLASS. If you omit it, the station uses the global segment 13 defined by STARTGEN, if any.

S13ID

Points to another storage class that defines a shared segment 13. This permits stations that must use different storage class definitions to use the same shared segment 13.

STARTGEN -- Start Controller Configuration

STARTGEN identifies and indicates the beginning of a controller configuration procedure and provides information about this procedure, the controller, and global segments 13 and 15.

STARTGEN must be the first configuration macro instruction in every controller configuration procedure and can appear only once.

Name	Operation	Operand
[label]	STARTGEN	ID=name,DATE=mmddy [,VN={n 0}] [,CTLR={4701-1 4701-2 4701-5}] [,S13={nnnn 0}] [,S15={nnnn 0}] [,MSGLITE={n 4}] [,DSKOP={Y N}] [,MONITOR= { (STD EXP CNM [,log] CNMBM EXPMB [,segs] [,log]) }] [,TOALERT={N Y}] [,TIMEOUT={m:ss 2:00}] [,PRINT={NOGEN GEN}] [,STATS={NOWRAP WRAP}] [,DUMP={N Y}] [,BSC={X'usc' C'usc' X'F1'}] [,CNM=(id, [term] [,tranpl])] [,TRACE={nn 50}]

ID

Specifies a name (1-8 alphameric characters) that identifies this controller configuration procedure. This configuration procedure is cataloged on the program library using this name, and the date specified by DATE.

DATE

Specifies the month, day, and year of this controller configuration procedure.

VN

Specifies the configuration version, in the range 0-255. The default value is 0.

CTLR

Is the machine and model number of the controller that is to use this configuration.

S13

Is a decimal number specifying the size of segment 13. The default value is 0. This value defines the global segment 13 that is shared by all stations that do not have a shared segment 13 associated with their storage class.

S15

Is a decimal number specifying the size of the optional portion of segment 15. The fixed portion of segment 15 (see the copy DEFGMS instruction) is not included in this specification. The default value is 0.

MSGLITE

Specifies which indicator light (1, 2, 3, or 4) on the control operator's terminal indicates a log message. The default is the CHECK light (4). You can also specify zero if you don't want a light, or a string of numbers (only as many as four) to set on multiple lights.

DSKOP

Specifies whether the diskette is operational (Y) or nonoperational (N). When you specify a diskette to be operational, the monitor's debugging functions that alter (patch) controller storage or modify (perform absolute diskette I/O on) diskette sectors are not permissible. These restricted monitor functions are denoted in *Subsystem Operating Procedures*. When monitor changes to the diskette and storage are permissible, a count field in the system control data set gets updated each time a change occurs.

TIMEOUT

Is a decimal number indicating the number of minutes and seconds that the controller is to wait before an automatic start occurs. The default value is 2 minutes. If you specify 0, an automatic start occurs immediately, and no entry from the keyboard is possible.

The type of startup is determined by (1) the value specified in a SETDSKT (code X'02') instruction, (2) the value entered by the control operator before a time-out occurs, (3) the value specified in a SETDSKT (code X'04') instruction if a time-out occurs, or (4) an automatic warm start (as though a 2 were entered in response to the 00001 message) if the time-out occurs and no SETDSKT instruction was issued.

TOALERT

If specified (Y) or allowed to default, puts an alert message in the system log when a terminal error occurs.

MONITOR

Specifies whether the standard (STD -- the default) or the expanded system monitor should be set up in this system. You specify an expanded system monitor as follows:

- Specify EXP for the expanded monitor when you **don't** want CNM active at IPL.
- Specify CNM for the expanded version with CNM/CS support active at IPL.
- Specify EXPMB or CNMBM for the particular version of the expanded system monitor that you need when you also need multiple-block capability.

Note: If you code either the CNM or CNMBM, or the EXP or EXPMB suboperands of the MONITOR operand (above), and you plan to use CNM/CS, you must also code the CNM operand as described below.

The suboperands *segs* and *log* are optional and positional. You can code a value **only** for *log* if you specify MONITOR= STD or EXP or CNM. You can code values for **both** *segs* and *log* if you specify MONITOR= EXPMB or CNMBM.

If you code values for either or both, you must enclose the entire expression following "MONITOR=" within parentheses. If you do not use either suboperand, you need not code the parentheses.

If you code values for both *segs* and *log*, you must do so in the order illustrated because they are positional. If you code a value for *log* without preceding it with a value for *segs*, you must code two commas. The first comma denotes the position for *segs*.

Use *segs* to specify the number of 256-byte diskette sectors that are used for a multiple-block system monitor. The greater the number of 256-byte sectors used, the faster the system monitor operates. The range is 2-26. The default is 15.

Use *log* to specify whether you want the occurrence of system monitor commands recorded in the system log. The valid values for *log* are either Y (for yes) or N (for no). N is the default.

If your system creates or updates the operating diskette using the host transmission facility, you can't use the standard monitor. You must instead specify any of the other monitor options. Additionally, your FILES macro must specify EDAM with the ALLOC option, and your OPTMOD macro must specify the P41 and P5E optional modules.

PRINT

Specifies whether a listing is to be produced to contain only the user input statements (NOGEN) or to contain the macro expansion statements (GEN). The default is GEN.

STATS

Specifies whether the statistical counters should wrap around to 128 when they have reached 255, or stop at 255 without wrapping. NOWRAP stops the counter at 255 without wrapping. WRAP allows the counters to wrap to 128, and is the default value. You should use WRAP if you use the *4700 Problem Determination Guide*, GC31-2033, to analyze system problems.

DUMP

Allows a choice of either dumping controller storage or reloading from the operating diskette after a system failure. If you specify Y or do not code the DUMP operand, the content of controller storage is dumped to a scratch diskette that you must insert in the primary diskette drive after the error occurs. Specify N if you want the controller to reload from the operating diskette after a system failure.

BSC

Changes the following BSC system monitor default parameter when COMLINK specifies TYPE=1422.

usc Specifies the one-byte or two-byte unit select and specific poll characters to be used for the system monitor. The default is X'F1'.

CNM

Specifies the CNM/CS environment, where:

id Specifies the station number to be used for CNM processing.

term Specifies (by the DEFADDR or DCAPORT label) the keyboard-display to be used for CNM station.

The *term* operand is optional and positional. If you do not specify a label for *term*, the CNM station does not have a keyboard-display assigned to it. Also, if you do not use *term*, but do use *tranpl*, which follows positionally, you must code the positional comma for *term*, for example:

```
CNM=(3,,POOL1)
```

Note: If the label is to refer to a DCAPORT macro, it must be preceded by a 'D/' in the *term* parameter. This limits the length of this DCAPORT label to six characters.

Example: The expression —

```
term=D/PORT01
```

— specifies that the keyboard-display device defined by a DCAPORT macro whose label is PORT01 is to be assigned to the CNM station.

tranpl Specifies (if used) the name of the user TRANPL statement to be used by the CNM station.

Note: These operands are equivalent to the ID and TERM parameters of the STATION macro, but they must have unique values. That is, you must code a value for *id*, and optionally for *term*, on STARTGEN if you code the CMN parameter, but the values you specify must not be the same as the values you use for the equivalent parameters on any STATION macro in this CPGEN.

The *tranpl* operand is optional. If you do not specify *tranpl*, a private general pool for the CNM station is the default.

TRACE

Optional parameter in which *nn* indicates the percentage of available storage that the diagnostic trace area is to have. “Available” storage consists of any storage that remains in the controller after load image processing is complete. When you specify the TRANPL macro in the CPGEN, this available storage must support both the diagnostic trace area and the storage areas for any transient application programs. The TRACE parameter designates what percentage the trace area is to have. The trace area gets all available storage when the CPGEN includes no TRANPL macro.

The default value is 50 (percent) if the CPGEN includes a TRANPL macro.

STATION -- Specify a Station

STATION gives the necessary information about a logical work station. You must code one STATION macro for every logical work station except station 1. Station 1 is reserved for use by the controller.

The STATION macro relates the actual device descriptions in it and the DEFADDR, DCAPORT, and LDA macros with the logical device addresses and the name of a controller application program.

Name	Operation	Operand
[label]	STATION	ID=nn ,APBNM={name (name[,vn 1])} ,DELSET=X'nn' [,TERM={term (term([lda,...][,A B]),...)}] [,DA=(port([lda,...])[,...])] [,MAXSTOR={n 0}] [,SS=class] [,CPU=(N Y[,PU] [,BIND] [,NOPURGE] [,NOAUTO])] [,SHARED={Y N}] [,STARTUP={Y N}] [,WRT={n 1}] [,INSTR={nnnn 5000}] [,ASYNLDA=(lda[,...]) [,BSC=({X'usc' C'usc'} [,DEF])] [,RETSTK={n 6}] [,INTMR={n 0}] [,DSM=(fctentries,row1,column1,row2,column2,EAB,DCA [,APSTACK={n 0}] [,MAXSEG=n] [,DPOOL=poolname] [,TRANPL=poollabel] [,DTACC={n 0}] [,SEGALL={n 0}]

ID

Specifies the identification number of this station (2-60).

APBNM

Specifies the name and version number of the controller application program associated with this station. The version number is the number specified in the BEGIN instruction for nonrelocatable programs, or it is the number specified in the APNAME parameter of Host Support's LINKAPB command. For COBOL programs, APBNM is the program name used in the PROGRAM-ID paragraph.

Note: Multiple versions of the same application program are not permissible in one configuration.

TERM

Identifies the list of terminals associated with this station:

term

Is the label of:

- The DEFADDR macro that defines the loop address of the device or devices to be assigned

- OR -

- The label of the LDA macro that describes:
 - The specific device of a multiple-device group

- OR -

- The specific component of a multiple-component device

If you specify an LDA label, the additional operands in TERM= do not apply.

lda

Is the logical device address or addresses of the device components to be assigned to this station. The default LDA assignment is as follows:

LDA Terminal

0	3614 or 3624 terminal
0	4704 or 3604 keyboard
0	3606 keyboard-display-magnetic stripe reader
0	3608 keyboard-display-magnetic stripe reader
1	4704 or 3604 display
1	3608 printer
2	4704 or 3604 magnetic stripe reader-encoder
3	4710 or 3610 or 3612 document printer
3	3616 journal printer
3	3615 administrative terminal printer
4	3611 or 3612 passbook printer
4	3616 document printer
6&7	(reserved)

The financial services terminals (3606 and 3608) are not supported by the local configuration facility.

Components in the same work station cannot have the same LDAs. If two components that are to be in the same work station are assigned to the same LDA by default (for example, the 4704 keyboard and the 3614 terminal), only one of the components can use that LDA. During the controller configuration procedure, the first of these components coded in the TERM operand of the STATION instruction can take the LDA assigned by default. The other component, however, must then be coded to include a specification for another LDA.

Similarly, if more than one 4710 or 3610 document printer, 3611 passbook printer, 3612 passbook and document printer, or 3616 printer are in the same logical work station, the terminals cannot all use the LDAs assigned by default. When the 3612 gets an LDA (either by default or because you assign one), that LDA is associated with the document printer, while the next consecutive LDA is automatically assigned to the passbook printer.

For example, if you assign the 3612 the LDA of 6, LDA 6 applies to the document printer, and LDA 7 to the passbook printer. Similarly, when the 3616 gets an LDA (either by default or because you assign it one), that LDA is associated with the journal printer. The next consecutive LDA is in turn associated automatically with the document printer.

If the number of logical device addresses specified is not sufficient to assign all components, the following assignments are made:

- For component devices that can have variable addresses, additional logical device addresses are generated by increasing from the last-specified logical device address. If you don't specify any logical device addresses, the default mentioned above then applies to the first component. All additional components then receive LDAs in consecutive order, increasing from the first (default) address.
- For component devices that cannot have variable addresses, the default values listed above take effect.

A or B

If this is a shared terminal, this operand specifies whether A or B is used as an indicator for this station. The A or B should be used for a single printer only.

If you code either A or B, the terminal is flagged as *shared*. The A or B indicator is valid only if the terminal is a 3610, 3611, 3612, 3616, 4710, or 4720.

DA

Specifies the devices attached through the device cluster adapter (DCA).

port

Is the label of a DCAPORT macro that defines a port to be associated with this station.

lda

Is the logical device address of the first component of a multiple-component device, or the logical device address for a single device. The default LDA assignment is as follows:

LDA Terminal

0	3278 or 3279 or 4704-2 keyboard
1	3278 or 3279 or 4704-2 display
6	3262 or 3287 printer

MAXSTOR

Defines the total storage that is required at any one time by the initial application program and any active subapplication programs. You must specify this value, which is a decimal number, if called application programs define their own storage requirements.

The total storage necessary is based on the initial application's segments (0, 2-12) plus the sub-application's segments (0, 2-13) that are not passed to it. Segment 13 for the initial application is not part of MAXSTOR.

If neither the configuration (using this parameter) nor the initial application program (using DEFSTOR) specifies the maximum storage, a called application program can use only the initial application program's storage. It can allocate none of its own. This parameter also applies to COBOL programs using subprogram linkage.

SS

Is a decimal number that specifies the class of storage (1-240) to be associated with this work station, or the total number of segments used by this work station. You describe a storage class by using the SEGSTOR macro. If this station's initial application program does not call other application programs, or if the called programs share the initial application program's storage, SS can define all of the necessary storage. Otherwise, the called programs contain their own storage definitions. If the SS value specifies the total segments used, do not code the MAXSEG, or MAXSTOR operands.

DELSET

Specifies the initial setting of the delimiter control byte.

CPU

Specifies whether this work station is to be able to communicate with the host processor (Y) or not (N). The default value is Y. PU specifies that this station should receive and send the System Network Architecture (SNA) network services commands REQMS and RECFMS.

BIND specifies that the station is to receive all BIND parameters, even though BIND is not specified in the COMLINK macro.

NOPURGE specifies that negative responses sent by the station to any part of a chained message do not cause the remainder of the chain to be purged. Instead, the station receives the remaining parts of the chain. If you do not specify this option on either the STATION or COMLINK macro, negative responses cause the purging of the remaining parts of chained messages so that the station will not receive them.

NOAUTO specifies that the controller is not to respond automatically to any SNA command other than ACTPU, DACTPU, ACTLU, or DACTLU. NOAUTO also specifies that the controller does not send an automatic response when a station issues an LEXIT instruction, but does send an automatic response when the station issues another LREAD instruction.

SHARED

Specifies whether this work station is to be shared between two operators (Y) or not (N). If you omit this operand or specify N, the station is not shared, and two segment 0's are not generated.

STARTUP

If you specify Y, the controller application program associated with this station receives control when the controller starts. If you omit this operand or specify N, this station does not receive control during startup. If you specify Y, you must also specify a startup entry point in the STP operand of the application program's BEGIN instruction.

WRT

Is the decimal number in the range 1-7 that indicates how many LWRITE instructions the host processor can receive before an automatic check of the sequence of write operations begins. The default value is 1.

Note: WRT must not be greater than the value of the WRT operand in the COMLINK macro instruction.

INSTR

Is a decimal number that indicates the number of instructions that can be executed between LEXIT instructions. Exceeding the count causes a program check 6 and prevents an unending loop from retaining control of the controller. The range of values that you can specify is 1-65535 (X'01'-X'FFFF'). The default value is 5000.

ASYNLDA

Is the decimal number (0-7) that indicates the logical device addresses (LDAs) that can cause the logical work station to be dispatched. These LDAs are the addresses of the 4704, 3604, 3606, 3608, 3278, or 3279 keyboards, or 3614 or 3624 Consumer Transaction Facilities assigned to the logical work station. Within parentheses, you can specify sequential LDAs by coding the lower LDA, a hyphen, and the higher LDA (for example, ASYNLDA=(0,3-5)). The default value for this operand is LDA 0.

BSC

Specifies that the BSC default parameters for this station are to be changed.

usc

Specifies the one-byte or two-byte unit-select and specific-poll characters to be used for this logical work station. The default values for stations 2-60 are:

<i>Station IDs:</i>	<i>Poll-Select Characters:</i>
2-9	X'F1'-X'F9'
10-18	X'C1'-X'C9'
19-27	X'D1'-X'D9'
28-35	X'E2'-X'E9'
36-44	X'81'-X'89'
45-53	X'91'-X'99'
54-60	X'A2'-X'A8'

DEF

Specifies that this is the default logical work station that is to receive all messages for this controller that did not contain a valid unit selection character. Only one logical work station can use this option in the controller.

RETSTK

Specifies the maximum number (0-255) of return addresses that the return-address stack can contain at one time. The return-address stack is at the end of the station's segment 1. The number of bytes reserved for the return addresses is equal to 2 times n. This operand affects the length of the station's segment 1 because that segment must be large enough to accommodate the return-address stack. If you do not specify RETSTK, the return-address stack can contain a maximum of 6 return addresses.

INTMR

Defines the number of interval timers to be set up for this work station. Specify zero (0) if you don't want any timers, or set n to the total number of timers you do want allocated. The maximum is 15.

DSM

You must code this operand if this station operates with the 3270 data stream mapping (DATSM) function. DSM defines the DATSM image buffer, alternative display, and field control table (FCT) sizes. It also specifies whether the extended attribute buffer should be allocated and how the output data stream represents erase orders.

The DATSM image buffer holds each original screen image. Each image comprises fields, and each field requires an entry in the field control table. You define the number of FCT entries needed by this station. To do this, select (from all screen images to be processed by this station) the original screen image with the most fields. Count the fields in this image, and specify this field count as *fcentries*.

To define the image buffer, specify the maximum screen size that the station is to use, where *row1* is the number of rows and *column1* is the number of columns. The values for *row2* and *column2* represent the initial original screen size, which DATSM uses when the application program does not specify an original screen size by using the PUTDMS instruction.

If this station processes DATSM color or APL data, specify EAB to allocate the extended attribute buffer. To replace erase orders in the output data stream with blanks for improved data flow, specify DCA.

DCA

Specifies that DATSM is to put blanks (rather than erase orders) in an output data stream to improve performance for DCA-attached devices.

EAB

Specifies the extended attribute buffer.

SBF

Specifies that the station is to use the system image buffer for DATSM transactions rather than one of its own. Refer also to the description for using the DEFSBF macro, elsewhere in this chapter.

APSTACK

Defines the maximum level of stacking, or nesting, of called application programs invoked by APCALL instructions. If no application programs are called, do not code APSTACK. If you specify 14 for *n* (the maximum), this station can have a total of 15 programs active at one time. If you do not code APSTACK, zero (0) is assumed, and an APCALL instruction in the initial application program causes a program check. This parameter also applies to COBOL programs using subprogram linkage. If the DTACC value is not zero, the maximum APSTACK value is 13.

MAXSEG

Specifies the maximum number of segments the work station requires at any one time. If you do not specify a segment count for MAXSEG, but APSTACK is specified, the default is determined as follows:

$$n = (\text{APSTACK} + 1) \times (16 + B)$$

— where *B* is 0 if SHARED=N, or 1 if SHARED=Y. Otherwise, the total number of segments is as specified by SS, or defaults to 16. This parameter also applies to COBOL programs using subprogram linkage.

DPOOL

Optional operand that associates a pool of devices with the station through the use of the symbolic name *poolname*. This name is used as an operand on the DCAPORT or DEFADDR macros to assign devices to a device pool.

TRANPL

Optional operand that identifies the storage pool (defined by the TRANPL macro) that is to be used by this station. Multiple STATION macros can access the same storage by specifying the appropriate TRANPL macro label.

The value for *poollabel* must be the same as the label coded on the TRANPL macro that defines the storage characteristics desired.

Warning: Be very careful how you set up the sharing of storage areas among work stations. The improper use or management of these areas by application programs can result in the suspension of a work station — or perhaps even the entire system — while it waits indefinitely for the availability of storage in a pool that cannot become available.

DTACC

Defines the number of modules that the station can hold simultaneously as a result of using DTACCESS. The default value is zero. The maximum value is 11.

SEGALL

Defines the number (*n*) of dynamic segments that the station can allocate for concurrent use before having to free segments in order to allocate more. The default value for *n* is 0, and the maximum value is 64.

TRANPL -- Define a Storage Pool

TRANPL defines a storage pool, which can be one of two types:

- A *station* pool
- A *segment-class* pool.

To define a station pool (a storage pool for the use of a specific work station), you code the label of the TRANPL macro that defines that pool on the TRANPL operand of the STATION macro for the station that is to have the use of the pool. Multiple stations can share the use of a single pool if the STATION macro for each specifies the same label on its TRANPL operand.

To define a segment-class pool, you code the ID operand of the TRANPL macro and do **not** refer to the label of this TRANPL macro from any STATION macros.

To activate the storage manager's capability for using "refreshable" storage, you can use the RFSH parameter either alone or in conjunction with other pool-definition parameters.

Name	Operation	Operand
label	TRANPL	$\left\{ \begin{array}{l} (\text{size}, \text{count}) \dots [, \text{RFSH} = \{ \text{Y} \text{N} \}] \\ \text{size} [, \text{ID} = n] [, \text{RFSH} = \text{Y} \text{N}] \\ \text{RFSH} = \text{Y} \text{N} \end{array} \right\}$

label

A symbolic name (1-8 alphanumeric characters) for this pool description. The TRANPL operand of one or more STATION macros refers to this label if the pool is to be a station pool, as described above.

size,count

Size is a decimal number that specifies the size of the storage areas in this pool. *Count* is a decimal number (1-255) that specifies the number of transient areas of that size to be allocated for this pool.

Note: This form of macro expression remains for Release 2 compatibility only. The form described below is preferable for Release 3.

size

The size of the pool for this storage area.

ID

The decimal identifying number (in the range 1-15) to be associated with this storage pool, if the pool is to be a segment-class pool.

If you do not specify an ID number, the pool becomes a station pool, so you must then code the label of this TRANPL macro expression on the TRANPL operand of one or more STATION macros.

RFSH

Specifies whether you want the refreshable-storage option active for this operational system. If you code Y, you enable the storage manager to reuse any read-only storage to satisfy a storage request for another station. In doing so, the storage manager reloads that storage from disk or diskette when references to it reoccur.

You can protect individual modules from this sort of reuse by coding **REFRESH=N** for them on the APOPT statement of their application program. Additional information about this option is in *Controller Programming Library — Volume 1*.

The default value for RFSH on the TRANPL macro is N.

TRTBHDR -- Specify Translation Table Headers

TRTBHDR specifies which of the shift cases, defined by INTRTBL macros, are to be used for a 3604, 3278, 3279, or 4704 keyboard. It also specifies which key is to be used as the reset key. When the configuration macros are assembled, the TRTBHDR macro must appear before the INTRTBL macros.

Name	Operation	Operand
label	TRTBHDR	INTRTBL=(lab1 , [lab2] , [lab3] [,lab4]) [,CANCEL={xx 00}] [,DKATBL=dkalabel] [,TYP A={ (xx [,xx , . . . ,xx) }] [,SHFT={ (xx [,xx , . . . ,xx) }]

label

Is a required label that corresponds to the label used in the TRTBHDR operand of the device's DEVxxxx configuration macro.

INTRTBL

Specifies the labels (as many as 4) from the INTRTBL macros that build the translation tables for each shift case. You must specify the label from the INTRTBL macro for shift case 1 or the keyboard is invalid. If you don't need any of the other cases, you can omit the corresponding INTRTBL labels, but you must still code the separating commas.

CANCEL

Specifies the scan code of the reset key for this keyboard. The scan code is specified as two hexadecimal digits (00-7F) with no other descriptors. The default value is 00.

DKATBL

dkalabel is the label of the user-specified dead-key accent table. Any symbolic name (1-8 alphameric characters) is valid except one that begins with the characters BUA, which are reserved.

CANCEL

Optional operand that specifies the scan code for the reset key for this keyboard. You specify the scan code as two hexadecimal digits in the range 00-7F.

TYP A

Specifies the scan codes for the repeat-action (typematic) keys for an alphameric keyboard (repeat-action keys cannot be specified for numeric keyboards). You specify each scan code as two hexadecimal digits (00-7F) with no other descriptors.

Note that you cannot use this function with a keyboard attached to the secondary port of a 4704 (that is, the 50-key keyboard when used with a 62-key keyboard to make a 112-key keyboard).

SHFT

Specifies the scan codes for the transient shift keys for an alphanumeric keyboard. (Note that some devices, such as certain 3604 models, do not support transient shift keys). You specify each scan code as two hexadecimal digits (00-7F) with no other descriptors. If a key is to be a transient shift key (that is, a shift key that puts the keyboard in the specified case only while you are pressing the key), you must also define it in the associated INTRTBL macros.

Note that you cannot use this function with a keyboard attached to the secondary port of a 4704 (that is, the 50-key keyboard when used with a 62-key keyboard to make a 112-key keyboard).

Notes:

1. The total number of scan codes specified in the TYP A and SHFT operands cannot exceed 16.
2. The same scan code cannot be specified by both TYP A and SHFT.
3. Do not specify scan code 00 and the scan code assigned as the reset key (CANCEL) for TYP A or SHFT.
4. Do not specify scan code 3F for 4704.

If the translation tables being defined are to be used with keyboards that have repeat-action (typematic) capability, but no repeat-action (TYP A) or downshift (SHFT) keys are to be defined, you should specify TYP A=7F to generate a null control table for the repeat-action facility.

TRTLIST -- Specify a Translation Table List

The TRTLIST macro defines data to be used with the DCATRTBL macro. You use this macro only if you intend to use the 4704-2 device in its local-keytracking mode.

Name	Operation	Operands
label	TRTLIST	SHIFT1=case1, SHIFT2=case2, SHIFT3=case3, SHIFT4=case4, ALTTBL=alt

label

Is the required symbolic name (1-8 alphameric characters) that the DCA4704 uses to identify this macro.

SHIFT1

case1 specifies the label of the DCATRTBL macro that describes the relationship between the lowercase keyboard scan code and the display's regen character code. If you omit this specification, the default for the device goes into effect.

SHIFT2

case2 specifies the label of the DCATRTBL macro that describes the relationship between the uppercase keyboard scan code and the display's regen character code. If you omit this specification, the default for the device goes into effect.

SHIFT3

case3 specifies the label of the DCATRTBL macro that describes the relationship between the lowercase keyboard scan code and the display's regen character code. If you omit this specification, the default for the device goes into effect.

SHIFT4

case4 specifies the label of the DCATRTBL macro that describes the relationship between the uppercase keyboard scan code and the display's regen character code. If you omit this specification, the default for the device goes into effect.

ALTTBL

alt specifies the label of the DCATRTBL macro that describes the relationship between the combination Alt key-and-keyboard scan code and the display's regen character code. If you omit this specification, the default for the device goes into effect.

Chapter 4. Using the Local Configuration Facility -- LCF

This chapter describes how to use the local configuration facility (LCF) at the IBM 4700 Finance Communication System to enter control program generation (CPGEN) information and to create an operating diskette. It includes both reference and guidance information describing how to use LCF.

LCF can be used by people with various levels of data processing experience. The CPGEN process that underlies LCF, however, is complex and requires programming experience. Thus, your institution might choose to have a programmer write a complete set of CPGEN statements for an operator to enter, and design work sheets to communicate that information to the operator.

Introducing LCF

LCF enables you to enter and maintain CPGEN information at your 4700 system rather than at the host system. When you have decided on the configuration of your IBM 4700 system, use LCF to enter the information needed to define that configuration to your controller.

Without LCF, you enter your configuration statements at the host system and assemble them as you would a program. When the assembly is complete, you analyze the output to detect any errors. You store the resulting CPGEN file at the host, and transmit copies to your controllers.

With LCF, you can design a CPGEN procedure and hand-deliver or transmit that procedure to each controller site. A control operator can enter the CPGEN information directly into the 4700 system from an attached display station. A host system is therefore not necessary.

LCF can operate conversationally: as you enter a CPGEN statement, LCF analyzes that statement for correct syntax before accepting the next statement. When LCF discovers an incorrect statement, it displays the statement along with an error message, and prompts for a replacement. LCF can also read previously-defined CPGEN statement files from diskettes or the host. When the CPGEN file is complete, you then use LCF to create an operating diskette.

CPGEN, What is it ?

CPGEN (control program generation) is a set of statements with which you identify and define each component of your system. These statements form a body of coded information that you use to define the characteristics of your 4700 devices to the controller. You must define the characteristics of the controller itself, the characteristics of each device, and the characteristics of each work station.

You must understand CPGEN thoroughly to be able to design the CPGEN macros that will be entered through LCF. The LCF operator, however, need not understand CPGEN.

Using LCF Procedures

LCF presents a series of menus describing the available functions. You select an item from the menu, and enter the necessary data.

To use LCF, you need the 4700 LCF program diskette. Then do the following things:

1. Be sure the power is switched on to the 4700 system and to your display station, and that the terminal's loop address switches are set for loop 1, address 1 — or, if you are using a 4704 DCA device, that it is connected to DCA port 0.
2. Attach an IBM 3615 Printer on loop 1, address 3 — or an IBM 3287 DCA printer to DCA port 1 — if you want to print the LCF statement file at the controller.
3. Insert the 4700 LCF program diskette and wait for the startup sequence to complete.

The LCF program diskette performs the same kind of startup procedure as does your operating diskette. During startup, you must include some optional modules when the 00001 prompting message appears. The optional modules you enter depend on the size of the controller's storage. Two versions of the LCF program diskette are available. One version is for controllers with 192K storage, and the other version is for controllers with 256K or more of storage. The section "Starting LCF" later in this chapter explains which optional modules to include for each size of controller.

You then select LCF from the LCF program diskette menu. When LCF displays its own menu, installation is complete.

Save the LCF pre-operating diskette. LCF will prompt you to insert it when you are ready to create your operating diskette, as this manual shows later.

LCF requires the use of an LCF library diskette to save the files of CPGEN source statements. The diskettes that you must furnish for use as LCF library diskettes must be 4700-formatted diskettes.

Creating the CPGEN File

Normally, you will use LCF to create a CPGEN file for later use in creating the operating diskette. For each CPGEN statement, LCF prompts you to enter a single 80-column card image. Use standard assembler-language conventions for 80-column card format, such as:

- Start all statement labels in column 1.
- Use columns 1-71 for your data.
- Use a non-blank character in column 72 to indicate that the statement is continued on a subsequent card image (known as a continuation card).
- Start a continuation card in column 16, leaving columns 1-15 blank.
- Start the symbolic name in column 1. We suggest that you start the CPGEN statement name in column 10, and the CPGEN operands after column 16.
- LCF ignores any user data in columns 73 through 80 and might use these columns when processing statements.

Editing the CPGEN Information

After you have entered CPGEN statements into LCF, you can use the LCF editor to add, change, and delete CPGEN statements. When the CPGEN information is complete and correct, you can save it on the library diskette.

Transferring Application Programs

You can use LCF to transfer assembled application programs from a 4700 diskette to the LCF pre-operating diskette.

Transferring an Entire CPGEN File

You can create an entire CPGEN file at a host system and transmit it to your 4700 system using either the host link or a basic exchange diskette. To use the host link you must write an application program in the host to build the data streams and to communicate with LCF. Then, use LCF to receive the CPGEN file and put it on a 4700 library diskette. You can transmit only the unassembled, source CPGEN file. LCF does not accept an assembled CPGEN file or a controller load image.

To use the basic exchange diskette, you write the CPGEN file on the diskette, and carry the diskette to the site of the 4700 controller. A basic exchange diskette is one that meets the standard format described in the *IBM Diskette General Information Manual*. (More information about basic exchange diskettes appears later in this chapter.)

Maintaining a CPGEN Library

With LCF, you can maintain several complete CPGEN statement files on one or more 4700-formatted library diskettes. When you are ready to create an operating diskette, you can choose a CPGEN statement file from the library diskette. You can also save the output of an LCF session on a library diskette for subsequent transmission to a host system using the host transmission facility.

Using LCF

A basic procedure for using LCF appears later in this chapter. This procedure can be performed by people of varying skills. An operator can use the procedure along with a list of CPGEN statements prepared by a programmer, to enter the statements and to create an operating diskette.

LCF's display panels, menus, and messages lead you through its operation. You can learn to use LCF by simply following the displayed instructions. The first few times, however, you should use the examples in this chapter to guide you through LCF. You might, for example, try to enter the CPGEN statements as shown in the example later in this chapter, making the same errors and correcting them in the same way. When you have mastered LCF, you might need only the reference information that describes the LCF panels and messages.

Before you use LCF, take a look (later in this chapter) at the list of limitations that govern the ability of LCF to process CPGEN statements.

Preparing an LCF Generation

The CPGEN programmer uses the 4700 programming books to decide how to select and code the necessary CPGEN statements. Then, the programmer can write the statements on a coding sheet, or other structured form, for entry by the LCF operator. Nothing else is necessary, because LCF is self-guiding, and this manual contains the reference material for the messages and panels. However, the programmer can also prepare a separate procedure showing which LCF functions to use to enter the CPGEN statements.

Checking for the LCF Prerequisites

To use LCF, you must have an IBM 4700 controller and an IBM 4704 Display screen and keyboard. (The smallest permissible screen size is 480 characters.) This display terminal must be attached to loop 1, address 1 of the controller, or — if you are using a 4704 device attached through a device cluster adapter (DCA) — it should be attached to DCA port 0. Before using LCF, read the *IBM 4704 Operating Procedures* for information on how to install and use the terminal, and the *IBM 4700 Subsystem Operating Procedures* for logon and startup procedures.

You must, of course, have the LCF program itself. LCF comes on two diskettes. The 4700 LCF program diskette contains the LCF functions, a copy of the expanded system monitor, and a startup procedure, as described in the *IBM 4700 Subsystem Operating Procedures*. The LCF pre-operating diskette contains the information for building the operating diskette.

Note: You ought to make several copies of this pre-operating diskette for backup.

50-Key Keyboard Support

You can use any 4704 keyboard with LCF. However, the 50-key keyboard does not have engraved keys. If you use the 50-key numeric/function keyboard, you will want to know how LCF interprets that keyboard. Figure 4-1 illustrates the 50-key keyboard with keys marked as LCF interprets them. You might want to type from this figure, use it to make a keyboard template, or attach keytop labels to mark the keys.

RESET	SHFT	←	→	: *	+ -	()	=	ENTER
1	2	3	4	5	6	7	8	9	0
Q	W	E	R	T	Y	U	I	O	P
A	S	D	F	G	H	J	K	L	'
Z	X	C	V	B	N	M	,	SPACE (X'40')	(NOT USED)

Figure 4-1. 50-Key Numeric/Function Keyboard as Interpreted by LCF

Optional Printer

You can also attach an IBM 3615 Printer to loop 1, address 3, or an IBM 3287 printer to DCA port 1. This printer is useful for printing the LCF statement file at the controller.

Library Diskette Support

The *library diskette* is a user-supplied and formatted diskette that LCF uses to save and restore CPGEN source files that you enter. You can keep a library of these source files on one or more library diskettes.

Using LCF

This topic contains a series of sample procedures to show you how to use LCF. (More detailed reference information about each LCF display panel appears later in this chapter.) Several of these sample procedures use CPGEN statements. You need not understand the CPGEN statements to use these sample procedures or to use LCF.

Before you begin, assemble the materials you will need:

- The LCF pre-operating diskette
- The 4700 LCF program diskette
- A 4700 controller
- An attached display monitor and keyboard

We recommend that you have several backup copies of your installation's pre-operating diskette. See the *4700 Subsystem Operating Procedures* for information describing how to copy diskettes.

If you will be moving a file or application program from another diskette, have that diskette available and marked clearly for identification. If you will be using CPGEN statement files from a basic exchange diskette or a library diskette, have that diskette marked and ready to insert.

Starting LCF

Switch on the 4700 controller and display station, and insert the 4700 LCF program diskette in the diskette drive as shown in the *IBM 4700 Subsystem Operating Procedures*. The 4700 performs its startup procedure and displays the 00001 message. Follow your installation's startup procedure. Two versions of the LCF program diskette are available. One version is for controllers with 192K storage, and the other version is for controllers with 256K or more of storage.

When you see the 00001 prompting message, the line on the screen above the 00001 will contain either LCFMN3 or LCFR30. If that line contains ...

- LCFMN3, you are working with the smaller (192K) controller.

Enter the optional modules 1B, 1D, 5E, 68, and 29.

- LCFR30, you are working with a 256K or larger controller.

Enter the optional modules 1B, 1D, 5E, 68, 29, B2, and B4.

Note: If a large CPGEN is to be processed, refer to "Maximum CPGEN File Size" later in this chapter for possible modifications to this step.

When startup is complete, the 4700 system displays the 4700 LCF diskette menu. Select LCF from this menu.

```

** 4700 LCF DISKETTE MENU **
( 1 ) ENTER LCF MODE
( 2 ) CHANGE PASSWORD
( 3 ) ENTER SYSTEM MONITOR
ENTER SELECTION NUMBER___
_
```

LCF then displays its own menu, with the cursor at the lower left of the screen as shown below:

```

                                LCF
SELECT:
(1) DEFINE CONFIGURATION
(2) EDIT CONFIGURATION
(3) CREATE OPERATING DISKETTE
(4) AP TRANSFER UTILITY
(5) RECEIVE CPGEN STATEMENTS
(6) RETURN TO MAIN MENU
-
                                nnn

```

Note: The *nnn* value at the lower-right corner of the screen denotes the maximum number of card images allowed by LCF.

Throughout this book this panel is referred to as the main LCF menu.

Creating the CPGEN File

This section shows you how to enter CPGEN statements from the keyboard. If this is your first time through, type 1 (option 1 on the menu), and press the Enter key. The following menu then appears:

```

                                STATEMENT FILE _____
SELECT:
(1) USE THIS FILE
(2) DISPLAY FILE NAMES
(3) GET ANOTHER FILE
(4) START A NEW FILE
(5) DELETE A FILE
(6) RETURN TO PREVIOUS MENU
-

```

Enter “4” to start a new statement file. LCF displays an empty statement-entry panel.

Note: The file name displayed at the top of the menu is the name of the currently active file — that is, the file in storage and available for use by LCF. When you first invoke LCF, no file is in memory, so the name that initially displays is UNNAMED, to indicate this first-time situation. As soon as you name a file (by saving it to a library diskette), or LCF reads in a new file because you select menu option 3, the new file name then displays.

```

                                ENTER STATEMENT
-----1-----*-----2-----3-----4
-
-----5-----6-----7-*-----8

```

Enter CPGEN
statement
here ----->

Type one 80-column card image into this panel. (The first statement in the CPGEN file is usually STARTGEN.)

The input panel shows you the positions for columns 10, 16, 20, 30, 40, 50, 60, 70, 72, and 80. (The asterisks indicate columns 16 and 72.) Use the standard assembler-language conventions for 80-column card format, such as:

- Start all statement labels in column 1.
- Use columns 1-71 for your data.
- Use a non-blank character in column 72 to indicate that the statement is continued on a subsequent card image (known as a continuation card).
- Start a continuation card in column 16, leaving columns 1-15 blank.
- Start the symbolic name in column 1. We suggest that you start the CPGEN statement name in column 10, and the CPGEN operands after column 16.
- LCF ignores any user data in columns 73 through 80, and might use these columns when processing statement files.

The display panel for your STARTGEN statement might appear like this:

```

      ENTER STATEMENT
-----1-----*---2-----3-----4|
|L1          STARTGEN  ID=TESTGEN1,DATE=050|
|781,MONITOR=EXP,S13=300|
-----5-----6-----7-*-----8|

```

Notice that the entire card image fits into two lines of the displayed panel. Column 72 is empty to indicate that this card ends this STARTGEN statement. LCF displays a complete 80-column card image in two lines of the panel to accommodate 80 columns in a 40-column screen.

Entering Comments

You can enter a comment on a separate card image, or as part of a CPGEN statement card image. To enter a comment card, type an asterisk (*) in column 1, and follow the asterisk with your comment.

ENTER STATEMENT							
-----1-----*	2-----3-----4						
* THIS IS A COMMENT							
-----5-----6-----7-*	-----8						

To enter a comment on the same panel with a CPGEN statement, omit the asterisk and leave at least one blank between the end of the statement and the comment:

ENTER STATEMENT							
-----1-----*	2-----3-----4						
L1 STARTGEN ID=TESTGEN1,DATE=050							
781,MONITOR=EXP,S13=300 COMMENT							
-----5-----6-----7-*	-----8						

Remember, columns 73 through 80 are ignored, and column 72 is used only to indicate continuation. So, any part of a comment that extends to column 72 indicates that the next card is a continuation card.

To file the statement, press Enter.

Correcting Errors

As you enter each statement (using option 1), LCF performs some preliminary error checking. If you have spelled a keyword incorrectly, omitted a required operand, used incorrect syntax, or selected an option that is not valid, LCF displays an error message and again displays the input panel. LCF displays the statement up to the error, but removes the entire statement from LCF storage. On the empty panel, enter the statement correctly. For example, enter this statement as part of your CPGEN input:

```
ENTER STATEMENT
|-----1-----*---2-----3-----4|
|D3604  DEV3604 MODEL=4 ,OUTTTBL=OUTTAB1|
| ,TRTBHDR=INTAB1                       |
|-----5-----6-----7-*-----8|
```

You have misspelled the OUTRTBL keyword. LCF displays this error message:

```
ENTER STATEMENT
|D3604 DEV3604 MODEL=4 ,OUTTTBL
|
|
|
|
| 17 INVALID KEYWORD
```

Note that the OUTRTBL keyword is misspelled. Remember the correct form of the statement so that you can reenter it correctly. Press the Enter key to erase the screen. LCF displays the blank input panel so that you can enter the statement correctly.

```
ENTER STATEMENT
|-----1-----*---2-----3-----4|
|D3604  DEV3604 MODEL=4 ,OUTRTBL=OUTTAB1|
| ,TRTBHDR=INTAB1                       |
|-----5-----6-----7-*-----8|
```

LCF adds the corrected statement to the CPGEN file when you press Enter.

Using Continuation Cards

With LCF you can continue CPGEN statements from one card image to another. To learn how to enter statements longer than 71 characters, use the STATION statement to try continuation panels. The STATION statement will be too long for one 80-column panel. Enter the first image, with a non-blank character in column 72:

```
ENTER STATEMENT
-----1-----*---2-----3-----4
      STATION ID=2,TERM=(DEF1),APNM=
FINAP,CPU=N,STARTUP=Y,          *
-----5-----6-----7-*-----8
```

The asterisk (*) in column 72 tells LCF that you will continue this statement on another panel. After you press the Enter key to enter this first display panel, LCF presents another input panel for the remainder of the statement. Begin the remainder of the statement in column 16, leaving columns 1-15 blank.

```
ENTER STATEMENT CONTINUATION
-----1-----*---2-----3-----4
              DELSET=X'FF',INSTR=65535
-----5-----6-----7-*-----8
```

Because you entered no character in column 72, LCF adds the completed STATION statement to the CPGEN file when you press Enter. Remember that if you enter any character in column 72, LCF treats the next panel as a continuation, and the continuation panel must start in column 16.

Ending CPGEN Entry

When you have entered all your CPGEN statements, press the Enter key without typing any data into the input panel. LCF, receiving no input, stops building the statement file, and asks you whether you want to save these CPGEN statements as a file on the library diskette.

Storing Files on the Library Diskette

The library diskette is a standard 4700-formatted diskette on which you can store several files of CPGEN statements and several completely processed configurations. When you end statement entry, or when you enter the S (Save) command from the LCF editor, LCF prompts:

```
SAVE TO LIB DSKT (Y/N)
```

Enter a Y if you want to save these CPGEN statements as a file on a library diskette. If you enter N, LCF does not store the CPGEN statements but retains them in storage as the active file. Then the main LCF menu reappears.

After you enter Y in response to the previous prompt, LCF asks you to insert the library diskette into either the primary or secondary drive (the primary drive is the drive unit used to IPL the controller; the secondary drive is then the other drive unit). Then LCF prompts you to:

```
146 ENTER NAME FOR STMT FILE
```

LCF first checks to be sure that the file name is not a duplicate. If the name does not already exist, LCF stores the file. If that file name is already on the library diskette, LCF prompts:

```
147 DATA SET EXISTS. REPLACE (Y/N)
```

If you reply N (you do not want to replace the file) LCF does not store this file on the library diskette. If you reply Y, LCF stores the new file on the library diskette.

In any case, when a file is stored on the library diskette, LCF displays this confirmation message:

```
_____ SAVED TO LIB DSKT
```

Editing the CPGEN File

After you enter your CPGEN information, use the LCF editor to review and change each statement. The LCF editor presents one card image (display panel) at a time for your inspection. You can add or delete panels, change the statements, move backward and forward in the file, and finally save the entire CPGEN file for creating the operating diskette.

Note: If you select option 2, the editing option, but you have not yet created a CPGEN statement file, LCF displays an error message and again displays the main LCF menu. You can use the editing option only when a CPGEN file already exists.

To use the editor, enter a 2 on the main LCF menu: The following menu will then appear:

```
STATEMENT FILE _____
SELECT:
(1) USE THIS FILE
(2) DISPLAY FILE NAMES
(3) GET ANOTHER FILE
(4) START A NEW FILE
(5) DELETE A FILE
(6) RETURN TO PREVIOUS MENU
```

To use the editor, enter a 1 to edit the file name displayed at the top of the menu.

Note: The file name displayed at the top of the menu is the name of the currently active file — that is, the file in storage and available for use by LCF. When you first invoke LCF, no file is in memory, so the name that initially displays is UNNAMED, to indicate this first-time situation. As soon as you name a file (by saving it to a library diskette), or LCF reads in a new file because you select menu option 3, the new file name then displays.

LCF then displays the EDIT panel, containing the first panel of the first CPGEN statement in the file. The first statement is usually STARTGEN.

```

                                EDIT
-----1-----*---2-----3-----4
Statement Area | L1          STARTGEN  ID=TESTGEN1,DATE=050
                | 781,MONITOR=EXP,S13=300
-----5-----6-----7-*-----8
Command Line   | _
Command Meanings
                | F  =FORWARD    C  =CHANGE    A=ADD
                | B  =BACKWARD   DEL=DELETE   S=SAVE
                | ST =SRCH TYPE  SL =SRCH LBL  E=EXIT
                | DUP=DUPLICATE SR =RPT SRCH  P=PRINT

```

Notice the edit commands at the bottom of the screen, and the position of the cursor. The cursor is at the command line.

The F command presents the next panel in the file. Use the F command to move through the CPGEN from front to back, one panel at a time. Or, enter F followed by a space and the number of panels to move forward. (If you enter a number greater than the file size, LCF displays the last panel in the file.)

```

                                EDIT
-----1-----*---2-----3-----4
Command Line   | L1          STARTGEN  ID=TESTGEN1,DATE=050
                | 781,MONITOR=EXP,S13=300
-----5-----6-----7-*-----8
Command Line   | _
Command Meanings
                | F  =FORWARD    C  =CHANGE    A=ADD
                | B  =BACKWARD   DEL=DELETE   S=SAVE
                | ST =SRCH TYPE  SL =SRCH LBL  E=EXIT
                | DUP=DUPLICATE SR =RPT SRCH  P=PRINT

```

The F 6 command will display the sixth panel after the current panel.

The C command moves the cursor under the first character of the displayed CPGEN line. You can then move the cursor, using the cursor-moving keys, to correct the statement. When you have corrected that panel, press Enter to move the cursor to the command line again, and enter S to save the change permanently, or F or B to move to another panel.

The A command inserts space in the CPGEN file for one panel, and presents an empty panel into which you can enter new information.

The B command presents the previous panel in the file. Use B to move backward through the file. As with the F command, you can follow B with a space and the number of panels to move backward through the file, stopping at the first if the number is too large.

The DEL command removes the displayed panel from the CPGEN file.

The S command saves changes permanently. If you enter S after changing a panel, that change is recorded permanently. If you enter S after changing several panels, those changes are then recorded permanently.

If you make several changes and neglect to enter the S command before leaving the editor, your changes are not recorded. You do get a second chance, however. LCF asks:

ARE YOU SURE (Y/N)

If you reply Y, LCF does not save your changes to the library diskette, but they do remain in storage as the active file. If you reply N, the EDIT panel reappears.

You can either enter the S command after each change or at the end, after all changes are made. LCF will ask you if you also want the statements stored on the library diskette. Before you exit the LCF editor, be sure that you enter the S command to make all your previous changes permanent.

The ST command searches the remaining CPGEN statements in the file for a specified statement type. If you enter ST DEFADDR, LCF displays the next DEFADDR statement (or a NOT FOUND message if no further DEFADDR statements exist).

The SL command searches the remaining CPGEN statements in the file for a specified symbolic name (statement label). If you enter SL LOOP1, LCF displays the next statement that has the label LOOP1 (or a NOT FOUND message if that label cannot be found).

The E command ends the LCF editor, and presents the main LCF menu so that you can select another LCF function. If the currently active file has been changed, you receive the following prompt:

ARE YOU SURE (Y/N)

The DUP command creates a duplicate of the current CPGEN statement, and files the duplicate immediately after the current statement. The duplicate (second iteration) becomes the current statement.

The SR command repeats execution of the previous ST or SL command. You can use SR to search for all occurrences of a specific statement type or statement label.

The P command prints the entire CPGEN file on the attached line printer.

Correcting a CPGEN Statement

Now invoke the LCF editor by entering a 2 in the main LCF menu. The following menu appears:

```
STATEMENT FILE _____
SELECT:
(1) USE THIS FILE
(2) DISPLAY FILE NAMES
(3) GET ANOTHER FILE
(4) START A NEW FILE
(5) DELETE A FILE
(6) RETURN TO PREVIOUS MENU
_
```

Enter a 1 to edit the file whose name appears at the top of the menu.

The LCF editor presents the first panel from the file. To change the date on your STARTGEN statement, enter the C command on the command line.

```
EDIT
-----1-----*---2-----3-----4|
L1      STARTGEN  ID=TESTGEN1,DATE=050|
781,MONITOR=EXP,S13=300
-----5-----6-----7-*-----8|
C_
F =FORWARD    C =CHANGE    A=ADD
B =BACKWARD   DEL=DELETE    S=SAVE
ST =SRCH TYPE SL =SRCH LBL  E=EXIT
DUP=DUPLICATE SR =RPT SRCH  P=PRINT
```

The editor moves the cursor beneath the first character of the STARTGEN statement and awaits your change.

```
EDIT
-----1-----*---2-----3-----4|
_L1     STARTGEN  ID=TESTGEN1,DATE=050|
781,MONITOR=EXP,S13=300
-----5-----6-----7-*-----8|
_
F =FORWARD    C =CHANGE    A=ADD
B =BACKWARD   DEL=DELETE    S=SAVE
ST =SRCH TYPE SL =SRCH LBL  E=EXIT
DUP=DUPLICATE SR =RPT SRCH  P=PRINT
```

Move the cursor under the characters you want to change, and key the changes (here, we change 050781 to 090981). Be sure to use the cursor-moving keys so that you do not inadvertently destroy any data. (DO NOT use the space key to move the cursor.) Press the Enter key to move the cursor to the command line.


```

          EDIT
-----1-----*---2-----3-----4
|L1      STARTGEN  ID=TESTGEN1,DATE=090
|981,MONITOR=EXP,S13=300
-----5-----6-----7-*-----8
|
| F  =FORWARD    C  =CHANGE    A=ADD
| B  =BACKWARD   DEL=DELETE    S=SAVE
| ST =SRCH TYPE  SL =SRCH LBL  E=EXIT
| DUP=DUPLICATE  SR =RPT SRCH  P=PRINT

```

Now, in order to see the next panel in the file, enter the F command to move forward one panel.

```

          EDIT
-----1-----*---2-----3-----4
|L1      STARTGEN  ID=TESTGEN1,DATE=090
|981,MONITOR=EXP,S13=300
-----5-----6-----7-*-----8
|F_
|
| F  =FORWARD    C  =CHANGE    A=ADD
| B  =BACKWARD   DEL=DELETE    S=SAVE
| ST =SRCH TYPE  SL =SRCH LBL  E=EXIT
| DUP=DUPLICATE  SR =RPT SRCH  P=PRINT

```

And LCF presents the next panel from the file:

```

          EDIT
-----1-----*---2-----3-----4
|L2      LOOPS    ID=1,TYPE=L,BPS=4800
-----5-----6-----7-*-----8
|
| F  =FORWARD    C  =CHANGE    A=ADD
| B  =BACKWARD   DEL=DELETE    S=SAVE
| ST =SRCH TYPE  SL =SRCH LBL  E=EXIT
| DUP=DUPLICATE  SR =RPT SRCH  P=PRINT

```

Adding a CPGEN Statement

Now add a FILES statement after the LOOPS statement. On the command line, enter the A command and press the Enter key.

```
EDIT
-----1-----*---2-----3-----4
L2          LOOPS  ID=1,TYPE=L,BPS=4800
-----5-----6-----7-*-----8
A
F  =FORWARD   C  =CHANGE   A=ADD
B  =BACKWARD  DEL=DELETE   S=SAVE
ST =SRCH TYPE SL =SRCH LBL E=EXIT
DUP=DUPLICATE SR =RPT SRCH P=PRINT
```

LCF displays an EDIT panel with no statement. The statement area is blank.

```
EDIT
-----1-----*---2-----3-----4
-
-----5-----6-----7-*-----8
F  =FORWARD   C  =CHANGE   A=ADD
B  =BACKWARD  DEL=DELETE   S=SAVE
ST =SRCH TYPE SL =SRCH LBL E=EXIT
DUP=DUPLICATE SR =RPT SRCH P=PRINT
```

Key your FILES statement in the input area, and press Enter to move the cursor back to the command line.

```
EDIT
-----1-----*---2-----3-----4
          FILES  PF=200,EDAM=( 2, 2,ALLOC )
-----5-----6-----7-*-----8
-
F  =FORWARD   C  =CHANGE   A=ADD
B  =BACKWARD  DEL=DELETE   S=SAVE
ST =SRCH TYPE SL =SRCH LBL E=EXIT
DUP=DUPLICATE SR =RPT SRCH P=PRINT
```

Deleting a CPGEN Statement

Use the DEL editing command to remove information from the CPGEN file one panel at a time. (Be sure you remove all panels of a CPGEN statement you no longer want.) Suppose LCF has presented the following panel. Enter the DEL command on the command line and press the Enter key.

```

          EDIT
-----1-----*---2-----3-----4|
|LB01  DEV3604   MODEL=4,OUTRTBL=LB12,TRT|
|BHDR=LB13,MSLITE=2,EOMSET=X'4F'|
-----5-----6-----7-*-----8|
|DEL_|
|
| F  =FORWARD   C  =CHANGE   A=ADD
| B  =BACKWARD  DEL=DELETE   S=SAVE
| ST =SRCH TYPE SL =SRCH LBL E=EXIT
| DUP=DUPLICATE SR =RPT SRCH P=PRINT
|

```

That panel no longer exists in the file. Then LCF presents the panel **following** the deleted panel.

Be careful when deleting panels from a continued statement. If you delete only the first panel, you might create a syntax error with an incomplete CPGEN statement.

Searching for Statements by Type

You can use the ST editing command to find a statement type (hence the ST) in the remainder of the file. Perhaps you want to find a LOOPS statement, so that you can add another loop to your configuration. On the command line, type the ST command followed by LOOPS.

```

          EDIT
-----1-----*---2-----3-----4|
|
|
-----5-----6-----7-*-----8|
|ST LOOPS|
|
| F  =FORWARD   C  =CHANGE   A=ADD
| B  =BACKWARD  DEL=DELETE   S=SAVE
| ST =SRCH TYPE SL =SRCH LBL E=EXIT
| DUP=DUPLICATE SR =RPT SRCH P=PRINT
|

```

And LCF displays the first LOOPS statement it finds.

```

          EDIT
-----1-----*---2-----3-----4|
|L2      LOOPS  ID=1,TYPE=L,BPS=4800|
|
-----5-----6-----7-*-----8|
|
|
| F  =FORWARD   C  =CHANGE   A=ADD
| B  =BACKWARD  DEL=DELETE   S=SAVE
| ST =SRCH TYPE SL =SRCH LBL E=EXIT
| DUP=DUPLICATE SR =RPT SRCH P=PRINT
|

```

Duplicating a CPGEN Statement

Use the DUP command to add another LOOPS statement just like this one.
Enter the DUP command.

```

                                EDIT
-----1-----*---2-----3-----4
|L2      LOOPS  ID=1,TYPE=L,BPS=4800
|
-----5-----6-----7-*-----8
|DUP_
|
| F  =FORWARD    C  =CHANGE    A=ADD
| B  =BACKWARD   DEL=DELETE    S=SAVE
| ST =SRCH TYPE  SL  =SRCH LBL  E=EXIT
| DUP=DUPLICATE  SR  =RPT SRCH  P=PRINT

```

Then enter the C command to change the duplicate LOOPS statement to a new LOOPS statement.

```

                                EDIT
-----1-----*---2-----3-----4
|L2      LOOPS  ID=1,TYPE=L,BPS=4800
|
-----5-----6-----7-*-----8
|C_
|
| F  =FORWARD    C  =CHANGE    A=ADD
| B  =BACKWARD   DEL=DELETE    S=SAVE
| ST =SRCH TYPE  SL  =SRCH LBL  E=EXIT
| DUP=DUPLICATE  SR  =RPT SRCH  P=PRINT

```

When LCF puts the cursor under the first character of the statement, use the cursor-moving keys to change the LOOPS statement to suit your needs. Finally, press the Enter key to move the cursor back to the command line.

```

                                EDIT
-----1-----*---2-----3-----4
|NEWLP   LOOPS  ID=2,TYPE=L,BPS=2400
|
-----5-----6-----7-*-----8
|
|
| F  =FORWARD    C  =CHANGE    A=ADD
| B  =BACKWARD   DEL=DELETE    S=SAVE
| ST =SRCH TYPE  SL  =SRCH LBL  E=EXIT
| DUP=DUPLICATE  SR  =RPT SRCH  P=PRINT

```

You have just added a LOOPS statement to define loop 2 in the CPGEN file.

Searching for a Statement Label

Suppose you have this statement on your screen:

```

                                EDIT
-----1-----*---2-----3-----4
| LB01  DEV3604  MODEL=4,OUTRTBL=LB12,TRT
| BHDR=LB13,MSLITE=2,EOMSET=X'4F'
-----5-----6-----7-*-----8
|
| F  =FORWARD   C  =CHANGE   A=ADD
| B  =BACKWARD  DEL=DELETE   S=SAVE
| ST =SRCH TYPE SL =SRCH LBL E=EXIT
| DUP=DUPLICATE SR =RPT SRCH P=PRINT

```

This DEV3604 macro has an OUTRTBL operand that refers to another CPGEN statement labeled LB12. To see what the statement labeled LB12 looks like, enter the SL command like this:

```

                                EDIT
-----1-----*---2-----3-----4
| LB01  DEV3604  MODEL=4,OUTRTBL=LB12,TRT
| BHDR=LB13,MSLITE=2,EOMSET=X'4F'
-----5-----6-----7-*-----8
| SL LB12
|
| F  =FORWARD   C  =CHANGE   A=ADD
| B  =BACKWARD  DEL=DELETE   S=SAVE
| ST =SRCH TYPE SL =SRCH LBL E=EXIT
| DUP=DUPLICATE SR =RPT SRCH P=PRINT

```

And LCF displays the statement with label LB12:

```

                                EDIT
-----1-----*---2-----3-----4
| LB12      OUTRTBL  DEVICE=3604,CHARSET=17
| 5
-----5-----6-----7-*-----8
|
| F  =FORWARD   C  =CHANGE   A=ADD
| B  =BACKWARD  DEL=DELETE   S=SAVE
| ST =SRCH TYPE SL =SRCH LBL E=EXIT
| DUP=DUPLICATE SR =RPT SRCH P=PRINT

```

Repeating Search Commands

Use the SR command to repeat the previous SL or ST command exactly. If you have just entered a:

```
ST DEFADDR
```

You can do it again by merely entering SR.

```

                                EDIT
-----1-----*---2-----3-----4|
NEWLP   LOOPS  ID=2,TYPE=L,BPS=2400  |
-----5-----6-----7-*-----8|
SR_
F  =FORWARD   C  =CHANGE   A=ADD
B  =BACKWARD  DEL=DELETE  S=SAVE
ST =SRCH TYPE SL =SRCH LBL E=EXIT
DUP=DUPLICATE SR =RPT SRCH P=PRINT

```

Printing the CPGEN Statement File

Use the P command to print the current CPGEN statement file on a 3615 or 3287 line printer attached to your controller.

```

                                EDIT
-----1-----*---2-----3-----4|
-----5-----6-----7-*-----8|
P_
F  =FORWARD   C  =CHANGE   A=ADD
B  =BACKWARD  DEL=DELETE  S=SAVE
ST =SRCH TYPE SL =SRCH LBL E=EXIT
DUP=DUPLICATE SR =RPT SRCH P=PRINT

```

LCF prints the CPGEN statement file on the line printer.

Leaving the LCF Editor

When you have performed all of your editing, enter the S command to save all changes in the permanent CPGEN file.

```

                                EDIT
-----1-----*-----2-----3-----4|
          FILES  PF=200,EDAM=(2,2,ALLOC)|
-----5-----6-----7-*-----8|
|S_|
|
| F  =FORWARD      C  =CHANGE      A=ADD
| B  =BACKWARD    DEL=DELETE      S=SAVE
| ST =SRCH TYPE   SL =SRCH LBL    E=EXIT
| DUP=DUPLICATE  SR =RPT SRCH    P=PRINT
|

```

LCF now displays the prompting sequence to see if you want to store the statement file on the library diskette as well as on the installation diskette. For information on this process, see "Storing Files on the Library Diskette" earlier in this chapter.

Finally, enter the E command to leave the editor. LCF then presents the main LCF menu. Usually, your next task will be to create an operating diskette from the CPGEN file you just created.

Creating an Operating Diskette

Now that you have created and edited a CPGEN file, you can create the operating diskette. You need both your 4700 LCF program diskette, and the LCF pre-operating diskette. LCF prompts for the appropriate diskette to be inserted at the required time on the required drive. (If you have one drive, the LCF program diskette is required first.) LCF first presents the main LCF menu. Enter selection 3 to create the operating diskette:

```

                                LCF
SELECT:
( 1 ) DEFINE CONFIGURATION
( 2 ) EDIT CONFIGURATION
( 3 ) CREATE OPERATING DISKETTE
( 4 ) AP TRANSFER UTILITY
( 5 ) RECEIVE CPGEN STATEMENTS
( 6 ) RETURN TO MAIN MENU
3                                nnn

```

```

                                STMT FILE _____
SELECT:
( 1 ) PROCESS CURRENT FILE
( 2 ) DISPLAY FILE NAMES
( 3 ) PROCESS A DIFFERENT FILE
( 4 ) DELETE A FILE
( 5 ) RETURN TO PREVIOUS MENU
-

```

To process the statement file name shown in the menu above, enter a 1. LCF then processes the file of CPGEN statements. As it processes each CPGEN statement, LCF briefly displays the statement verb (STARTGEN or LOOPS, for example), the statement label, and for a DEVxxxx or DCAxxxx macro, the label of the corresponding DEFADDR or DCAPORT macro.

```

L1D1      DEV3604  DEFADDR = L1T1

```

When it finds a statement that is not valid, LCF stops processing and displays an error message, the statement in error, and the sequence number of the statement in error.

```

91 STATEMENT NUMBER 012

-----1-----*---2-----3-----4
L1D1      DEV360471 MODEL=4,OUTRTBL=OUT36
047,OPTIONS=4905
-----5-----6-----7-*-----8

51 SYNTAX ERROR IN STMT FILE

```

Use the LCF editor to correct the error, and select this procedure again.

At any time during the creation of the operating diskette, you can cancel the entire operation. Press the Reset key twice. LCF displays this message:

```

88 COMPILE IN PROGRESS
ENTER C TO CANCEL,
DEPRESS ENTER TO CONT

```

If you really want to cancel the job, type a C and press the Enter key. If you want the job to continue, merely press the Enter key.

After processing all statements in the file, LCF then displays the message:

```

151 SAVE CONFIG DS TO LIB DSKT (Y/N)

```

(Enter,N if you are following this tutorial.) Following your response to this message, LCF then displays one of the following messages:

```

70 MOUNT PD DSKT ON PRIMARY

```

or

```

71 MOUNT PD DSKT ON SECONDARY

```

(PD means pre-operating. DSKT means diskette.)

Insert the LCF pre-operating diskette on the indicated drive, or replace the LCF program diskette with the pre-operating diskette on a single-drive system.

After the pre-operating diskette is mounted, you will receive one more chance to end the job before the pre-operating diskette is actually modified:

```
88 COMPILE IN PROGRESS
ENTER C TO CANCEL,
DEPRESS ENTER TO CONT.
```

Finally, LCF asks you whether you want unused optional modules deleted from the pre-operating diskette:

```
128 DELETE UNNEEDED OPTMODS
(Del TO DELETE, K TO KEEP ALL OPTMODS)
```

If you tell LCF to delete unused modules, keep backup copies of the pre-operating diskette. You might need those modules later if you update the configuration to include additional devices or functions.

If you have a single-drive system and removed the LCF program diskette previously, LCF asks you to:

```
97 MOUNT LCF DSKT ON PRIMARY
```

LCF again displays the main LCF menu.

After you create an operating diskette using LCF, you can reuse that diskette (when its contents are no longer needed), by using it as a pre-operating diskette.

Transferring Programs via Diskette

You can use LCF to transfer user-written application programs (called APs) from a diskette to the pre-operating diskette. Use the main LCF menu to enter option 4.

```

                                LCF
SELECT:
    ( 1 ) DEFINE CONFIGURATION
    ( 2 ) EDIT CONFIGURATION
    ( 3 ) CREATE OPERATING DISKETTE
    ( 4 ) AP TRANSFER UTILITY
    ( 5 ) RECEIVE CPGEN STATEMENTS
    ( 6 ) RETURN TO MAIN MENU
 4
                                nnn

```

LCF then displays this menu of transfer options:

```

                                LCF AP TRANSFER
SELECT
    ( 1 ) DEFINE AP SPACE
    ( 2 ) TRANSFER ALL APS
    ( 3 ) REPLACE SELECTED AP
    ( 4 ) ADD SELECTED AP
    ( 5 ) SHOW SYSAP SPACE
    ( 6 ) RETURN TO PREVIOUS MENU
-

```

These options are:

1. Allocate space on the pre-operating diskette. You must allocate space before actually transferring programs.
2. Transfer all application programs from the input diskette to the pre-operating diskette.
3. Replace a program, selected by the program name, on the pre-operating diskette with a program of the same name from the input diskette.
4. Add a program, by name, to those already on the pre-operating diskette.
5. Show the space for the system application program.
6. Leave this function. Display the main LCF menu.

Whenever you select option 2, 3, or 4, LCF first ensures that sufficient space exists and has been allocated (by option 1).

Allocating Space for Application Programs

Before you actually transfer programs to the pre-operating diskette, allocate sufficient space in which to store those programs. Select option 5 from the AP TRANSFER panel:

```
LCF AP TRANSFER

SELECT

( 1 ) DEFINE AP SPACE
( 2 ) TRANSFER ALL APS
( 3 ) REPLACE SELECTED AP
( 4 ) ADD SELECTED AP
( 5 ) SHOW SYSAP SPACE
( 6 ) RETURN TO PREVIOUS MENU

5
```

LCF now prompts you to:

72 MOUNT AP DISKETTE ON PRIMARY

or

125 MOUNT AP DISKETTE ON SECONDARY

Insert the correct diskette. Then LCF displays the following:

SYSAP SECTORS = *nn*

where *nn* is the total number of sectors necessary to contain all of the application programs.

APS IN SYSAP = *mm*

where *mm* is the total number of application programs on the application program diskette.

Make a note of the numbers returned by the messages above. They will be used in following steps. Now press Enter, and LCF displays:

PRESS ENTER FOR AP NAMES, C TO CANCEL

If you plan to transfer all of the application programs on the AP diskette, press C. If you plan to transfer only some of the APs, you can press Enter to get information about each individual AP on the AP diskette. For example, if you plan to transfer only one AP, you can find out how many sectors it requires and allocate only that much space in the following steps.

After determining the number of necessary sectors, press Enter to return to the AP Transfer menu.

```

LCF AP TRANSFER

SELECT

( 1 ) DEFINE AP SPACE
( 2 ) TRANSFER ALL APS
( 3 ) REPLACE SELECTED AP
( 4 ) ADD SELECTED AP
( 5 ) SHOW SYSAP SPACE
( 6 ) RETURN TO PREVIOUS MENU

  1

```

LCF now prompts you to:

```
ENTER NR. OF SECTORS
```

Enter the number of sectors necessary to contain the directory and application programs that will eventually be transferred. To reserve space for programs coming later, you can enter a number larger than you now need.

The number of sectors necessary for the directory depends on the total number of application programs for which you want to reserve space. The first 15 application programs require one additional sector. Application programs 16-31 require another additional sector, and so on.

For example, if you want to reserve space for 34 application programs, the necessary directory space will be:

<u>Application Programs</u>	<u>Directory Space (Sectors)</u>
1-15	1
16-31	1
32-34	+ 1

	Total = 3 sectors

LCF then prompts you to enter the maximum number of application programs that can be transferred. If you are transferring all programs, you must find out how many programs are on the diskette. If you do not indicate a number of programs, you will be able to transfer only one program.

```
ENTER MAX NR. OF APS
```

Enter the maximum number of application programs that might be transferred. Again, you can use this to reserve space for programs that will be transferred later.

Then, LCF prompts you to insert the pre-operating diskette:

```
70 MOUNT PD DSKT ON PRIMARY
```

Insert the correct diskette. LCF allocates the space.

Warning: Option 1 erases any application programs on the pre-operating diskette before it allocates the space. Before you use option 1 to allocate space, ensure that space is necessary. Perhaps you have already allocated sufficient space on the diskette.

Transferring All Application Programs

After you allocate space on the pre-operating diskette, select option 2 to transfer all application programs from the input diskette to the pre-operating diskette.

```
LCF AP TRANSFER
SELECT
(1) DEFINE AP SPACE
(2) TRANSFER ALL APS
(3) REPLACE SELECTED AP
(4) ADD SELECTED AP
(5) SHOW SYSAP SPACE
(6) RETURN TO PREVIOUS MENU
2
```

If you have one diskette drive, LCF first prompts you to insert the input (AP) diskette, then the pre-operating (PD) diskette. If there are many programs to transfer, LCF might again prompt you to insert the input diskette followed by the pre-operating diskette. Each time, LCF reads programs from one diskette and writes them on the other diskette.

When LCF completes, it has transferred all application programs from the input diskette to the pre-operating diskette.

Note that you can transfer application programs only from a 4700-formatted diskette to your 4700 system.

Replacing Application Programs

Use LCF to replace specific application programs on the pre-operating diskette with a program from the input diskette.

```
LCF AP TRANSFER
SELECT
( 1 ) DEFINE AP SPACE
( 2 ) TRANSFER ALL APS
( 3 ) REPLACE SELECTED AP
( 4 ) ADD SELECTED AP
( 5 ) SHOW SYSAP SPACE
( 6 ) RETURN TO PREVIOUS MENU
3
```

LCF first prompts you to insert the AP diskette and to enter the name of the program to be added; it displays:

```
125 MOUNT AP DSKT ON SECONDARY
```

LCF then prompts you to enter the name of the AP (application program) to be transferred:

```
99 ENTER AP NAME
```

Enter the name of the application program to be transferred, in one of the following ways:

name[R | T]

Causes the applications program you name to be transferred as either a transient (T) or resident (R) application program. The default is that the program retains its R or T characteristic from the source diskette. Refer to the description of the APCALL function in *4700 Controller Programming Library, Volume 1* for additional information regarding transient and resident application programs.

name*[R | T]

Causes all of the applications whose prefix is *name* to be transferred as either transient (T) or resident (R) application programs. The default is that the program retains its R or T characteristic from the source diskette. Refer to the description of the APCALL function in *4700 Controller Programming Library, Volume 1* for additional information regarding transient and resident application programs.

LCF transfers the program to the pre-operating diskette, replacing the program of the same name. If the program does already exist, LCF gives you one more chance to change your mind about replacing the program:

```
101 AP EXISTS - ENTER R TO REPLACE
```

If you know that you want to replace this program, enter R. (Otherwise, press Enter to cancel the operation.)

If the application program is not on the input (AP) diskette, or if the new program is larger than the old program and does not fit, LCF displays an error message and ends the procedure.

Adding Application Programs

Use LCF to add a program, by name, from any 4700-formatted diskette to your LCF pre-operating diskette.

```
LCF AP TRANSFER
SELECT
(1) DEFINE AP SPACE
(2) TRANSFER ALL APS
(3) REPLACE SELECTED AP
(4) ADD SELECTED AP
(5) SHOW SYSAP SPACE
(6) RETURN TO PREVIOUS MENU

4
```

LCF first prompts you to insert the AP diskette and to enter the name of the program to be added; it displays:

```
125 MOUNT AP DSKT ON SECONDARY
```

followed by

```
99 ENTER AP NAME
```

and

```
70 MOUNT PD DSKT ON PRIMARY
```

Insert the diskettes in the sequence requested by LCF, and enter the name of the program to be added.

name[R | T]

Causes the applications program you name to be transferred as either a transient (T) or resident (R) application program. The default is that the program retains its R or T characteristic from the source diskette. Refer to the description of the APCALL function in *4700 Controller Programming Library, Volume 1* for additional information regarding transient and resident application programs.

name*[R | T]

Causes all of the applications whose prefix is *name* to be transferred as either transient (T) or resident (R) application programs. The default is that the program retains its R or T characteristic from the source diskette. Refer to the description of the APCALL function in *4700 Controller Programming Library, Volume 1* for additional information regarding transient and resident application programs.

If there is enough space, LCF adds the program from the input diskette to the pre-operating diskette.

If the program already exists, LCF gives you one more chance to change your mind about replacing the program:

```
101 AP EXISTS - ENTER R TO REPLACE
```


If you are certain that you want to replace this program, enter R. (Otherwise, press Enter to cancel the operation.)

If the application program is not on the input (AP) diskette, or if the new program is larger than the old program and will not fit, LCF displays one of the following error messages and ends the procedure:

100 AP NOT FOUND

or

102 AP TOO LARGE TO REPLACE

Press the Enter key to return to the AP TRANSFER menu. See Appendix B for an explanation of the error message you receive.

Receiving a CPGEN File

You can use LCF's Option 5 to receive a complete CPGEN file from another system using either a diskette or the telecommunication network. (Earlier, under "Storing Files on the Library Diskette" you learned how to store the CPGEN file on the library diskette.) In either case, the entire CPGEN file is transferred from its incoming medium to LCF storage. You can now use LCF's Option 2 (EDIT) to save the CPGEN file to a library diskette. (Then, when you select LCF option 3, that CPGEN file is used to create the operating diskette.) From the main LCF menu, select option 5:

```
LCF
SELECT:
( 1) DEFINE CONFIGURATION
( 2) EDIT CONFIGURATION
( 3) CREATE OPERATING DISKETTE
( 4) AP TRANSFER UTILITY
( 5) RECEIVE CPGEN STATEMENTS
( 6) RETURN TO MAIN MENU
5                                nnn
```

LCF next prompts you to indicate whether to transfer the CPGEN file from a diskette or from the host link:

```
LCF RECEIVE CPGEN STATEMENTS
SELECT:
( 1) RECEIVE STMTS FROM HOST
( 2) RECEIVE STMTS BX DSKT
( 3) RETURN TO PREVIOUS MENU
-
```

If the CPGEN statements are to come from the host system by data link, enter a 1. If they are on a basic exchange diskette, enter a 2. Enter a 3 to return to the previous menu.

Receiving CPGEN Files on a Basic Exchange Diskette

LCF reads the CPGEN file from a basic exchange diskette into storage. LCF prompts you to insert the basic exchange diskette. On a system with one diskette drive, LCF prompts:

```
74 MOUNT BASIC EXCHANGE DSKT ON PRIMARY
```

On a system with two diskette drives, LCF prompts:

```
75 MOUNT BASIC EXCHANGE DSKT ON SECONDARY
```

After you insert the diskette, LCF prompts you to enter the name of the data set that contains the CPGEN file:

```
76 ENTER DATASET NAME
```

Enter the name of the particular CPGEN data set that is on the mounted basic exchange diskette. LCF reads the selected CPGEN file into storage.

When the process is complete, LCF presents the main LCF menu. You can now use LCF option 2 (EDIT) to save the file to an LCF library diskette.

Receiving CPGEN Files through the Host Link

LCF copies the CPGEN file from an attached host system using the telecommunication network. To use this option, your 4700 system must be attached to a host system. (Descriptions of the host-communication protocols and formats that an application in the host system must observe appear later in this chapter.) When finished, LCF displays the main LCF menu. You can now use LCF option 2 (EDIT) to save the file to an LCF library diskette.

Using LCF Panels

This section presents each LCF display panel, describes its format and purpose, and explains the meaning of each possible response or entry.

Using the 4700 LCF Program Menu

When you first invoke LCF, the menu below appears. LCF also displays this panel when you complete certain LCF operations. Use this menu to tell LCF what you want to do.

```

** 4700 LCF DISKETTE MENU **
( 1 ) ENTER LCF MODE
( 2 ) CHANGE PASSWORD
( 3 ) ENTER SYSTEM MONITOR
ENTER SELECTION NUMBER____
_
```

Selecting Option 1 permits you to perform tasks using the functions of the local configuration facility.

Changing the LCF Password

Selecting Option 2 enables you to change the LCF password, which protects the 4700 LCF program diskette. When you first receive a diskette, its password is 12345. You can change this password to be any string of 1-16 characters entered at the keyboard. When you select Option 2, above, you get this message:

```
ENTER NEW PASSWORD
```

When you get this message, enter the password you want to be in effect. You then get the following message:

```
ENTER OLD PASSWORD FOR VALIDATION
```

Entering the old password is a security precaution that ensures that, because you know the old password, you are authorized to select a new one. The new password is now in effect for all following logons to the LCF program diskette. Then you get the following message to indicate that the change is in effect:

```
NEW PASSWORD NOW IN EFFECT
PRESS ENTER TO CONTINUE
```

Pressing the Enter key with a null entry causes a return to the 4700 main LCF menu.

If you make an error in changing the password, you get the following message:

```
OLD PASSWORD INVALID
RE-ENTER
```

You can now either reenter the correct password or enter a null to return to the 4700 menu. After three invalid attempts, the system cancels the change-password function and returns you to the 4700 menu. The new password does not take effect until the message telling that it has appears.

Invoking the System Monitor

Selecting Option 3 enables you to invoke the system monitor. When you “log on” to the system monitor, the functions of the LCF program diskette give the monitor control of the work station. After you log off from the system monitor, you return control to the LCF program diskette by pressing the Enter key.

When you select the system monitor, the following message appears:

```
TO ENTER SYSTEM MONITOR
PRESS RESET KEY 3 TIMES
(PRESS ENTER KEY TO RETURN TO MAIN MENU)
```

You can now log on the system monitor. Additional information is in *4700 Subsystem Operating Procedures*. After you log off the system monitor, you can return to the 4700 LCF program menu by pressing the Enter key.

Note: System monitor commands that require the use of a printer can use only a 3615 on Loop 1, Address 3, or a DCA printer on DCA Port 1.

Using the Main LCF Menu

When you select Option 1 above, the main LCF menu appears, as shown below:

```

                                     LCF
SELECT:
( 1) DEFINE CONFIGURATION
( 2) EDIT CONFIGURATION
( 3) CREATE OPERATING DISKETTE
( 4) AP TRANSFER UTILITY
( 5) RECEIVE CPGEN STATEMENTS
( 6) RETURN TO MAIN MENU
-                                     nnn
```

Note: The *nnn* value at the lower-right corner of the screen denotes the maximum number of card images allowed by LCF.

The options and what they do are:

1. Create a CPGEN file using the CPGEN statements you enter from your terminal. LCF lets you add to or replace an existing CPGEN file, or create one for the first time.
2. Change an existing CPGEN file using the LCF editor. LCF displays one panel (card image) at a time, beginning with the first.
3. Create an operating diskette from a pre-operating diskette, using all of the CPGEN statements that you have entered, or a file that you have transferred to LCF (and perhaps modified using LCF).
4. Transfer one or more user-written application programs from a 4700-formatted diskette to LCF for inclusion on the operating diskette.
5. Receive a complete CPGEN file, in 80-column card format, from either the host system or a basic exchange diskette. LCF first asks you to identify the source of the CPGEN file.

- Return to the main menu on the LCF program diskette (the menu from which you selected LCF).

Defining a Configuration Panel

To define a configuration, select Option 1 from the main LCF menu. Then the following menu appears.

```
      STMT FILE
      SELECT:
      ( 1 ) PROCESS CURRENT FILE
      ( 2 ) DISPLAY FILE NAMES
      ( 3 ) PROCESS A DIFFERENT FILE
      ( 4 ) DELETE A FILE
      ( 5 ) RETURN TO PREVIOUS MENU
      -
```

If you select Option 1 from this menu, the “Statement Entry” panel shown below appears.

Statement Entry Panel

Use this panel to enter CPGEN statements into the file.

```
      ENTER STATEMENT
      -----1-----*-----2-----3-----4
      -
      -----5-----6-----7-*-----8
```

LCF displays this panel after you select option 1 to replace or add a panel in a CPGEN file.

The numbers within the panel indicate the column in units of 10 (2 indicates column 20, for example). The * symbols indicate columns 16 (to begin a continuation card), and 72 (to indicate continuation).

Enter an 80-column card image portion (71 data characters) of a statement on each panel, pressing the Enter key to store one panel and begin another. When you have no further panels to enter, press Enter while an empty panel is displayed.

Continuation Panel

LCF displays this panel after you put a non-blank character in column 72 of the preceding statement entry panel. (LCF processes the statement only after you enter the complete statement, that is, when you enter a panel with no character in column 72.)

ENTER STATEMENT CONTINUATION							
1	*	2	3	4			
5		6	7	*	8		

Use this panel to enter the next part of a continued statement or comment. The continuation data must start at column 16 (indicated by an * symbol). Columns 1-15 must be blank.

Edit Panel

LCF displays this panel when you select the LCF editor.

EDIT							
Statement Area	1	*	2	3	4		
Command Line	5		6	7	*	8	
Command Meanings	F	=FORWARD	C	=CHANGE	A	=ADD	
	B	=BACKWARD	DEL	=DELETE	S	=SAVE	
	ST	=SRCH TYPE	SL	=SRCH LBL	E	=EXIT	
	DUP	=DUPLICATE	SR	=RPT SRCH	P	=PRINT	

Usually, you will see a CPGEN statement in the statement area. When you first select the editor, LCF displays the first panel in the CPGEN file. You then go forward through the file one image at a time. You can enter the editing commands on the command line to perform these editing procedures:

- F** Display the next sequential 80-column card image panel in the CPGEN file. Follow F with a blank and a number to move forward that number of panels.
- C** Move the cursor from the command line to the statement area to key changes to the displayed panel.
- A** Display an empty statement area to add a panel at this point in the CPGEN file.
- B** Display the previous panel in the CPGEN file. Follow B with a blank and a number to move backward that number of panels.
- DEL** Delete this panel from the CPGEN file.

- S** Save all previous changes and additions to a library diskette. If you fail to enter the S command before you leave the LCF editor, none of your changes are made. You can enter S after each change, or enter it once when you have made your last change.

- ST** Follow the ST with the name of a CPGEN macro instruction. LCF searches the CPGEN file, beginning at the card image that follows the current card image, for a matching CPGEN statement. For example, if you enter ST DEFADDR, then LCF displays the next occurrence of the DEFADDR macro in your CPGEN file.

When LCF finds a matching card, it treats that card as the current card image. If LCF does not find a matching card image, LCF displays a NOT FOUND message. (The search ends at the end of the file. LCF does not begin again at the start of the file.)

- SL** Follow the SL with an alphameric symbolic name (a label) from a CPGEN statement. LCF searches the CPGEN file for that label, beginning with the next card image. LCF, on finding that label, displays the card image and treats it as the current card image. If it can't find the label, LCF displays a NOT FOUND message. (The search ends at the end of the CPGEN file. LCF does not begin again at the start of the file.)

- E** Leave the editor. Display the main LCF menu. LCF first asks you if you are sure that you want to leave the editor. You reply Y to leave the editor, or N to continue using it.

- DUP** DUP copies the current card image, and inserts the copy immediately after the current image. The new card (the duplicate) becomes the current card image.

- SR** SR repeats the last search you requested. If you have used ST or SL previously, you can use SR to repeat that search. This allows you to repeat a search without retyping the entire command.

- P** Prints the entire CPGEN file on an attached 3615 line printer or 3287 DCA printer.

Transfer AP Panel

LCF displays this panel when you select option 4 on the main LCF menu to transfer application programs from a 4700-formatted diskette to the LCF pre-operating diskette.

```

|                                     |
|                                     | LCF AP TRANSFER |
|                                     |
| SELECT                             |
|                                     |
|      ( 1 ) DEFINE AP SPACE         |
|      ( 2 ) TRANSFER ALL APS        |
|      ( 3 ) REPLACE SELECTED AP    |
|      ( 4 ) ADD SELECTED AP        |
|      ( 5 ) SHOW AP SPACE          |
|      ( 6 ) RETURN TO PREVIOUS MENU |
|                                     |
|      -                             |
|                                     |

```


You first define space on the installation diskette for the programs. Then you can transfer some or all applications programs (APs) from the input diskette to the pre-operating diskette. The options are:

1. Define the number of sectors and the number of programs. LCF prompts you to enter the number of sectors occupied by the programs to be transferred, and the number of programs.
2. Transfer all application programs from the input diskette to the pre-operating diskette. The process stops if LCF runs out of space (you did not allocate enough space using option 1).
3. Replace a program on the pre-operating diskette with a program of the same name from the input diskette. LCF prompts you for the application program name.
4. Add a program from the input diskette to the pre-operating diskette. LCF prompts you for the name of the program to be added.
5. Show the space requirements for each application program.

Receive CPGEN File Panel

Use this menu to tell LCF whether the CPGEN file will be transferred from the host by the telecommunication network, from the library diskette, or from a basic exchange diskette.

```
LCF RECEIVE CPGEN STATEMENTS
SELECT:
(1) RECEIVE STMTS FROM HOST
(2) RECEIVE STMTS FROM BX DSKT
(3) RETURN TO MAIN LCF MENU
—
```

LCF displays this menu after you enter a 5 in the main LCF menu to transfer a CPGEN file to the installation diskette. The options are:

1. LCF communicates with a user-written application program at the host computer to transfer a file of 80-column CPGEN card images from the host to the 4700 system. If LCF cannot find the file, it displays the RECEIVE CPGEN STATEMENTS file again.
2. LCF prompts you to insert a basic exchange diskette containing the CPGEN file. LCF transfers the file from that diskette to the installation diskette.
3. Leave this option. Return to the previous menu.

LCF Interfaces and Limitations

This section describes the various programming and data interfaces to LCF, as well as the limitations governing the use of LCF.

LCF Interfaces

Using LCF, you can transfer a complete CPGEN file from another 4700 system or from the host system on a diskette or through the telecommunication network. You can transfer 4700 application programs from a diskette to the 4700 operating diskette. This section describes the format you must use for each of these transfers.

CPGEN File from Diskette

You can use LCF to transfer a complete CPGEN file from a basic exchange diskette to your 4700 operating diskette.

The incoming diskette and CPGEN file must conform to these specifications:

- The incoming diskette, and the data sets on the diskette, must conform to the basic exchange diskette format as described in the *IBM Diskette General Information Manual*. Each data set on the diskette must be described in the directory in the diskette header.
- Each CPGEN statement must be in 80-column card format, stored in the first 80 bytes of a 128-byte diskette record. (LCF ignores the last 48 bytes of each record.)
- The last card image must be all blank (the first 80 bytes of the last record must contain blanks).

Transferring CPGEN Files from the Host

This section presents some general considerations for writing a host application program to send CPGEN statement files to a 4700 controller, and describes the transmission sequences that you should use.

Writing a Host Application Program: When writing the host application program, consider these factors:

- Access method characteristics
- Program requirements
- Operating procedures

Access Method Characteristics: You can use VTAM as the host telecommunications access method. The VTAM control blocks necessary for communication with a subsystem controller include the ACB, the NIB, and one or more RPLs.

For details about VTAM macro usage and program organization, refer to *ACF/VTAM Macro Reference* and *ACF/VTAM Macro Language Guide*. Transmit the host diskette image in two parts:

- Send the header 1 record as the only PIU in chain, requiring a positive response (+DR1).
- Upon receipt of a positive response, the host transmits the remainder of the image, starting with the header 2 record, as a single chain that requires only exception responses until last-in-chain. If exception responses do occur, transmission ends.

Use the following message termination after transmitting the CPGEN file:

1. LCF sends a positive response to the last-in-chain.
2. The host sends X'0588'.
3. LCF sends a positive response and then X'0581'.
4. The host sends an Unbind.
5. The 4700 sends a positive (+DR1) response.

Program Requirements: The host application should include the following steps in the sequence shown:

1. Initialize the files and control blocks.
2. Establish the session.
3. Transmit the CPGEN file.
4. End the session.

Operating Procedure: The system operator starts the telecommunication access method at the host, and then advises the control operator at the 4700 (usually by telephone) to start LCF and ensure that LCF is waiting for host contact.

The control operator then runs LCF and selects option 5 to receive the statement file from the host.

Then, the system operator starts the user-written host application program, defined to the access method as SYSSDC. This should establish a session with LCF.

Transmission Protocols

Use the following SNA (Systems Network Architecture) protocols to transfer complete CPGEN files from a user-written application program in the host system to LCF.

Initial Sequence

```
HOST                                     4700
-----
          ----- BIND ----->
<----- +DR1 ----- Response
          ----- SDT ----->
<----- +DR1 ----- Response
0103    ----- SINGLE WRITE ----->
<----- +DR1 ----- Response
```

Chaining Sequence

```
HOST                                     4700
-----
          ----- FIRST -----> CPGEN
          ----- MIDDLE -----> statements.
          .                       One 80-byte
          .                       record sent
          .                       in a 256-byte
          .                       record
          .                       for each
          .                       card in the
          .                       statement
          ----- LAST -----> file.
<----- +DR1 ----- Response
```

Termination Sequence

```
HOST                                     4700
-----
0588...  ----- SINGLE WRITE ----->
<----- +DR1 ----- Response
<----- SINGLE WRITE ----- 0581
Response ----- +DR1 ----->
----- UNBIND ----->
<----- +DR1 ----- Response
```

Error Ending

```
HOST                                     4700
-----
<----- -DR1 ----- 4 bytes of
                    status
<----- TERMINATE ----- Bind Resource ID
Response ----- +DR1 ----->
----- UNBIND ----->
<----- +DR1 -----
```

Parameters on Command Request Unit

Byte Description

0	Command Code - 01
1	Option Code - 03 (CPGEN statement file)

Transferring Programs by Diskette

Use a standard diskette in the 4700 format to transfer application programs from one system to another. The programs must reside in a data set named SYSAP, a direct data set having a physical sector length of 256 bytes. The first records of this data set must contain a directory of SYSAP. Space is available for 15 descriptor records in the first directory record, and 16 additional descriptor records on each subsequent directory record. The format for the directory follows.

The first 16 bytes of the first (or only) directory record are:

Byte	Content
0-1	Number of programs in SYSAP
2-3	Last relative record number of last program in SYSAP
4-5	Total directory slots (1 per program)
6-7	First relative record number of first program
8-15	Reserved

Each program is described by a 16-byte entry.

Byte	Content
0-7	Program name in EBCDIC, padded with X'40'
8	Program version
9	X'80' if another entry follows X'40' if transient application program
10-11	First relative record number of program
12-13	Last relative record number of program
14-15	Reserved

You can subtract the first relative record number from the last relative record number and then add 1 to the result to find the size (in records) of the program being transferred.

Maximum CPGEN File Sizes

This table shows the maximum number of CPGEN card images that you can have using LCF on 192K or 256K controllers. The column on the left shows the host attachment method. (If you are not using a host link to transmit a CPGEN file, set the controller switches to BSC to achieve the maximum capacity.)

When the LCF main menu appears, the maximum number of CPGEN card images is in the lower-right corner of the screen.

Attachment Method	Max File Size (Note 1)	Max File Size (Note 2)	Max File Size (Note 3)
BSC (L03)	410	455	635
HPCA/SNA (L01/L82)	405	450	630
HPCA/X21B (L01/L84)	340	385	565
HPCA/Multi-Use Loop (L01/L85)	385	430	610

Notes:

1. This maximum file size is the number of CPGEN card images in a single file supported by LCF when all optional modules on the 192K LCF program diskette are loaded in a 192K controller. Note that this results from entering no optional modules in response to message 00001, and it allows the use of a 3615 printer.
2. This maximum file size is the number of CPGEN card images in a single file supported by the 192K LCF program diskette when only the optional modules 1B, 1D, 5E, and 68 are loaded in a 192K controller. In this case, no printer is available for use by LCF.

3. This maximum file size is the number of CPGEN card images in a single file supported by LCF when all optional modules on the 256K LCF program diskette are loaded in a 256K controller. This allows the use of all of the I/O devices that LCF supports.
4. On controllers with 320K or more of storage, the maximum file size is 812 CPGEN card images.
5. If you do not require DCA devices, but do need a 3615 printer, use optional modules 1B, 1D, 29, 5E, 68, 8A, B2, and B4.

LCF Limitations

The following limitations affect either the LCF operator or the programmer responsible for preparing the LCF script.

Reserved Labels: Do not define symbolic names (statement labels) beginning with the characters BUA. BUxxxxx labels are reserved for system use. When coding translation tables, or SEGINIT or ENDGEN, do not allow fields enclosed in parentheses to span two panels.

This:

```

|-----1-----*---2-----3-----4|
|label  INTRTBL .....|
Panel 1 | (3F,C'F' ),(22,C'G' ),          * |
|-----5-----6-----7-*-----8|

|-----1-----*---2-----3-----4|
|          (23,C'X' ),KYBD=4662 |
Panel 2 |          |
|-----5-----6-----7-*-----8|

```

Not this:

```

|-----1-----*---2-----3-----4|
|label  INTRTBL .....|
Panel 1 | .....(3F,CF' ),(22,* |
|-----5-----6-----7-*-----8|

|-----1-----*---2-----3-----4|
|          C'G' ),(23,C'X' ),KYBD=4662 |
Panel 2 |          |
|-----5-----6-----7-*-----8|

```

INTRTBL, OUTRTBL, DCATRTBL, OUTSPEC, DEFADDR, CHARDEF, DCAPORT, MSTRTBL, and PINTBL: All these statements have positional operands that must precede any of the keyword operands.

This:

```
Label  INTRTBL  (3F,C'BAL' ),KYBD=4664,CASE=1
```

Not This:

```
Label  INTRTBL  KYBD=4664,CASE=1,(3F,C'BAL' )
```

| **DEFADDR:** You cannot use multiple subaddresses with DEFADDR. Examples of the forms supported by LCF appear below:

```
LABEL DEFADDR D1,ADDR=(L1,1,8)
LABEL DEFADDR (D1,2),ADDR=(L1,1,8)
LABEL DEFADDR D1,D2,ADDR=(L1,1,8)
LABEL DEFADDR D1,(D2,3),ADDR=(L1,1,8)
LABEL DEFADDR D1,(D2,3),(D3,4),ADDR=(L1,1,8)
```

CPGEN Statements not Supported: LCF rejects these CPGEN statements:

```
DEV3606 EXTCTR APMEDIA
DEV3608 PRIDSP OPTLIST
```

Assembler Control Statements: LCF does not use assembler control statements such as PRINT, TITLE, END, SPACE, and EJECT. Do not enter them while creating a CPGEN file using LCF option 1. However, LCF ignores these control statements when they are in a CPGEN file received from the host via the network or a diskette.

Segment Data: When segment data is specified in a SEGINIT or ENDGEN statement, only the following forms of data are recognized by LCF:

```
dddt'xxxx..x'
tlnn'xxxx..x'
ddd is a decimal number indicating the number
of times the constant is to be generated
(not required if one constant is to
be generated).
t is the notation describing the data between
the single quotation marks (X for hex, C for
character. Other types are NOT supported by
LCF). LCF does not allow you to enter a
single quotation mark (') in a character
string. If you want to include a single
quotation mark, use the hexadecimal
equivalent.
nn is a decimal number indicating the actual
length of the constant.
xxx.x is the data that makes up the constant.
```

INTRTBL Macro: On the first operand of the INTRTBL macro, code a comma if you omit the middle (character string) value.

```
INTRTBL (3C, ,CAS2)
```

Mounting Diskettes: When you insert a diskette, LCF might not respond immediately. During this time, LCF does not display a confirmation message and might seem to have forgotten you. Be patient. LCF is making the diskette ready and will be with you soon.

SEGINIT and ENDGEN Parameters: If the first parameter within parentheses is omitted on SEGINIT or ENDGEN, you must include the leading comma.

You must put SEGINIT statements into groups for each segment-and-station combination. Specify the displacement on the first statement of the group. Do not allow the system to assign a default value. On subsequent statements in the group, you can allow the system to use the default value.

Using the Library Diskette Statement File

The following topics describe various aspects of using library diskette statement files.

Creating Library Diskette Statement Files

You can create library diskette statement files after you finish entering statements (in LCF Option 1), or after you issue a S (Save) command while using LCF Option 2 (EDIT). The following prompt appears:

```
SAVE TO LIB DSKT (Y/N)
```

If you reply N (no), the statement file is not written to the library diskette. If you reply Y (yes), the statement file is written to the library diskette. If it is necessary to mount the library diskette, you then get the following prompt:

```
MOUNT LIB DSKT ON PRIMARY
```

```
or
```

```
MOUNT LIB DSKT ON SECONDARY
```

You then get a prompt to supply a name for the statement file to be saved:

```
ENTER NAME FOR STMT FILE
```

If LCF finds a data set with the same name on the library diskette, it issues you the following message:

```
DATA SET EXISTS. REPLACE (Y/N)
```

If you reply N, nothing is written to the library diskette. If you reply Y, LCF allocates the data set on the library diskette and writes the statement file to the newly-created data set. When the writing operation is complete, you get the following message:

```
DATA SET _____ ADDED TO LIB DSKT
```

Creating Configuration Data Set Files

The following prompt appears at the end of creating an LCF operating diskette (LCF Option 3):

```
SAVE TO LIB DSKT (Y/N)
```

If you reply N, processing continues. If you reply Y, you get the following prompt if the library diskette is not already mounted:

```
MOUNT LIB DSKT ON PRIMARY
```

```
OR
```

```
MOUNT LIB DSKT ON SECONDARY
```

LCF assigns a name to the configuration data set. It then checks to see if that name is already on the library diskette. If so, you get the following prompt:

```
DATA SET EXISTS. REPLACE (Y/N)
```

If you reply N, the data set is not replaced. If you reply Y, or if no name conflict existed, LCF writes the configuration data set to the library diskette and displays the following message:

```
DATA SET _____ ADDED TO LIB DSKT
```

LCF assigns a name to the configuration data set as follows: It uses the CPGEN name (from the STARTGEN statement) and appends a file type of LCF, separated from the CPGEN name by a period. For example, a CPGEN name of CONFIG1 produces a configuration data set named CONFIG1.LCF on the library diskette.

| Migrating from Release 1.0 to Subsequent Releases

| Release 1.0 of LCF saved CPGEN statement files on the installation diskette's permanent file. Subsequent releases of LCF, however, resides on the LCF program diskette and use the library diskette in place of the permanent file.

You can copy your existing configurations from the permanent file of a Release 1.0 installation diskette to a library diskette by using the COPY DATASET utility on the installation diskette. Use SYSPF as the name of the "from" data set and a name of your choosing for the "to" data set.

The resulting new data set on the library diskette will at first be as large as the permanent file on the installation diskette. This is often bigger than necessary because a statement file does not typically occupy the entire permanent file space. It is easy to compress the new data set, however.

| To do so, read the new data set using the LCF EDIT option. Make a change to the data set, and then save it. This operation reallocates the data set to its proper size, thus recovering the waste space on the library diskette.

Appendix A. CPGEN Messages

This appendix contains the CPGEN messages — error messages that can result from invoking the configuration macro instructions during the configuration procedure. The messages are in numeric sequence, along with explanations and recommended actions.

General CPGEN Messages

BDK502I STARTGEN MACRO NOT CODED

Explanation: STARTGEN is required to be the first macro for the configuration procedure.

System Action: The configuration procedure ends.

User Response: Code STARTGEN and reassemble.

BDK503I INVALID CPGEN ID

Explanation: The ID operand specified in the STARTGEN macro is longer than eight characters.

System Action: The configuration procedure ends.

User Response: Correct the ID operand and reassemble.

BDK506I INCORRECT DATE SPECIFICATION

Explanation: The DATE operand specified is longer than six digits or is nonnumeric.

System Action: The configuration procedure ends.

User Response: Correct the DATE operand and reassemble.

BDK507I MULTIPLE CALL TO STARTGEN

Explanation: STARTGEN can be issued only once in each configuration procedure.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK508I INVALID STORAGE CLASS SPECIFICATION

Explanation: The class specification is not within the range of 1 to 255 or it is nonnumeric.

System Action: The configuration procedure ends.

User Response: Correct the class specification and reassemble.

BDK509I INVALID STATION ID

Explanation: The station ID is either nonnumeric or is not in the range 2-60.

System Action: The configuration procedure ends.

User Response: Correct the ID and reassemble.

BDK510I DUPLICATE STATION ID

Explanation: The same ID has been assigned to another station; station IDs must be unique.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK512I WARNING: DUPLICATE SCAN CODE SPECIFICATION LAST SPECIFICATION TAKES PRECEDENCE

Explanation: The scan code number specified on the INTRTBL macro is the same for two or more different keys. The last key value specified with this scan code is accepted as an override of the previous specifications.

System Action: The configuration procedure completes, but the result is unpredictable.

User Response: Correct the error and reassemble.

**BDK513I WARNING: STATION (id); OPERAND WRT > COMLINK
OPERAND WRT: STATION WRT CHANGED TO COMLINK
WRT**

Explanation: The maximum number of consecutive write instructions issued from any station cannot exceed the buffer capacity specified for the link in the COMLINK macro.

System Action: The WRT operand on the STATION macro defaults to the maximum number allowed, and the configuration procedure completes.

User Response: Correct the error if necessary and reassemble.

**BDK514I WARNING: MAXIMUM LINK TIMEOUTS EXCEEDED:
DEFAULTED TO 85 MIN**

Explanation: The length specified for a link time-out exceeds the limit allowed; the length defaults to the maximum time allowed.

System Action: The configuration procedure completes with the length of the time-out set at 85 minutes.

User Response: Correct the error if necessary and reassemble.

BDK515I APBNM NAME INVALID

Explanation: The APBNM operand coded on the STATION macro is longer than eight characters.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK516I DEFADDR OR LDA ON THE STATION'S TERM=KEYWORD IS
INVALID**

Explanation: The DEFADDR or LDA label is longer than eight characters.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK518I INVALID DEVICE TYPE SPECIFICATION

Explanation: The device type specification in the DEV/DCAxxxx macro is invalid.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK523I SEGINIT MAY ONLY BE CODED FOLLOWING ENDGEN

Explanation: Self-explanatory.

System Action: Configuration ends.

User Response: If segment initialization is required, code SEGINIT after the ENDGEN macro.

BDK524I WARNING: MAXSEG VALUE LT SEGSTOR SEGMENTS FOR (station ID). DEFAULT TAKEN.

Explanation: The number of segments specified by MAXSEG=on the STATION macro is less than required by the initial application program for the station.

System Action: The number of segments required by the initial application program is allocated.

User Response: None.

BDK525I WARNING: MAXSTOR VALUE LT SEGSTOR SIZES FOR (station ID). DEFAULT TAKEN.

Explanation: The total storage specified by MAXSTOR for the station is less than that specified for the initial application program.

System Action: The storage required for the initial application program is allocated as the maximum at IPL.

User Response: None.

BDK526I WARNING: INVALID (description) FOR 4700 SYSTEM. (Action taken).

Explanation: This warning is for invalid functions, keywords, parameters, and so on. The invalid item might have been valid on another or earlier system or configuration.

System Action: Defaults are taken, as described in the “action taken” portion of the message.

User Response: None.

BDK527I WARNING: SHARED DESIGNATION IN CONFLICT (DEFADDR label)

Explanation: There is a conflict between the DEVxxxx macro and the STATION macro associated with the DEFFADDR label in the error message.

System Action: The configuration procedure completes, but the result is unpredictable.

User Response: Correct the error and reassemble.

BDK528I DUPLICATE DEFADDR OR LDA LABEL

Explanation: The same label has been used for two DEFADDR or LDA macros.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK529I APLIST MACRO -- (error description)

Explanation: The APLIST macro specified an invalid condition; for example, an invalid version number.

System Action: If a warning only, the configuration continues. Otherwise, the configuration procedure ends.

User Response: Correct the error and reassemble the configuration.

BDK530I INVALID CLASS SPECIFICATION

Explanation: The class specification on the STATION macro is either nonnumeric or not within the allowed range.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK531I INVALID SEGMENT SIZE SPECIFICATION

Explanation: The segment size specification is nonnumeric or not within the allowed range of values.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK532I INVALID SYSTEM LOOP SPECIFICATION

Explanation: A specification for Loop 1 is not acceptable for the system loop (Loop 1).

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK534I LOOP TYPE INVALID

Explanation: The TYPE operand in the LOOPS macro is not specified as L or R.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK535I LOOP NUMBER INVALID

Explanation: The loop number is either nonnumeric or not within the allowed range.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK536I DUPLICATE LOOP SPECIFICATION

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK537I DUPLICATE TLA

Explanation: Two components on a specified loop have been assigned the same terminal address.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK538I INVALID BUFFER LENGTH SPECIFICATION

Explanation: The buffer length specification is less than 40 or is nonnumeric.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK539I INVALID BUFFER NUMBER SPECIFICATION

Explanation: The buffer number specification is either nonnumeric or not within the permissible range.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK540I DUPLICATE USE OF COMLINK MACRO

Explanation: COMLINK can be specified only once.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK541I FILES MACRO MAY ONLY BE USED ONCE

Explanation: A FILES macro for a specific drive can be coded only once.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK542I WARNING: USER LOG TIME STAMP MAY ONLY BE SPECIFIED ON DRIVE ONE, IGNORED.

Explanation: The LOGTM keyword on the FILES macro can only be specified when DKT=1.

System Action: LOGTM is ignored.

User Response: None.

BDK543I WARNING: EOM WITH ZERO MASK IS INVALID

Explanation: Self-explanatory.

System Action: The configuration procedure completes, but the result is unpredictable.

User Response: Correct the error and reassemble.

BDK544I TOO MANY TRANSIENT FILES SPECIFIED

Explanation: The number of temporary files specified in the TF operand in the FILES macro is greater than the allowed maximum.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK547I NUMBER OF TRANSIENT FILES MISSING

Explanation: A subfile index is specified, but no temporary files were specified in the TF operand in the FILES macro.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK548I INVALID SUBFILE SPECIFICATION

Explanation: The subfile specification is not within the range 1-60.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK549I TOO MANY FILE INDEXES

Explanation: The subfile index coded and the number of temporary files specified are not in agreement.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK550I INCORRECT KEYWORD OPTION (keyword)

Explanation: The indicated keyword contains an incorrect option.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK551I REQUIRED KEYWORD OPERAND (keyword) OMITTED

Explanation: The indicated keyword is required, but has been omitted.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK553I INSTRUCTION LOOP THRESHOLD EXCEEDED

Explanation: The specified instruction loop threshold is greater than the maximum allowed.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK555I CONFLICTING LDA ASSIGNMENTS, STATION ID (id), LDA (lda)

Explanation: An LDA for the specified station conflicts with another LDA assignment.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK556I STATION ID (id) PARAMETER TERM REFERS TO AN UNDEFINED LDA, DEFADDR OR TERMINAL MACRO (label)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the label spelling, or define the omitted device, or remove the label from the STATION, TERM parameter if it is not necessary. Then reassemble.

**BDK557I WARNING: TOTAL (what) SPEED (number) EXCEEDS
(maximum) ALLOWED (with)**

Explanation: The total of the values of the BPS operands on all LOOPS macros exceeds 16,800, or the loop total plus the DCL operand on the COMLINK macro exceeds the maximum allowed for the configuration for the 4701-1, or exceeds 19,200 for the 4701-2.

System Action: The configuration procedure completes, but overruns might occur.

User Response: Correct the error and reassemble.

BDK558I CPGEN TERMINATED DUE TO ERRORS

Explanation: One or more serious errors have been detected.

System Action: The configuration procedure ends.

User Response: Correct the error(s) and reassemble.

BDK559I NO KEYBOARD/DISPLAY DEFINED

Explanation: This is an information message only.

System Action: The configuration procedure continues.

User Response: None.

BDK560I SEGMENT SPECIFICATION LIMIT EXCEEDED

Explanation: Only segments 0, 2 through 13, and 15 can be specified.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK562I WARNING: MORE THAN (count) 4800 BPS LOOP HAS BEEN
DEFINED**

Explanation: Only two 4800-bps loops can be specified with a BSC3 host link.

System Action: The configuration procedure completes, but response times might be slower.

User Response: Correct the error and reassemble.

BDK563I DUPLICATE SCAN CODE KEYWORD (keyword)

Explanation: The same scan code has been defined twice.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK565I IF THE FOLLOWING DS STATEMENT(S) HAS A LENGTH ERROR, THEN A CPGEN DATA AREA(S) HAS BEEN EXCEEDED

Explanation: One or more of the CPGEN data areas has exceeded its system-assigned limit, causing an assembler length error message in the form:

*** ERROR **** LENGTH ERROR -- L(number-(L'@@name-

where:

number=the number assigned to the data area by the system.

name=the name of the data area that has been exceeded:

@@COMME — for the common link area

@@DEV1E — for the loop-DCA area

@@DEV2E — for the diskette area

@@STATE — for the station area

To determine the number of bytes by which the area was exceeded, calculate as follows:

$((@@xxxxE-@@xxxxB-number) \times 4096) + \text{address-value}$

with the values above obtained from the assembler's cross-reference area:

@@xxxxB = the symbol's length value

@@xxxxE = the symbol's length value

'number' = from the error message, defined above

address-value = the three low-order hex values of symbol @@xxxx, where "xxxx" is COMM, DEV1, DEV2, or STAT.

System Action: The assembly completes, but the CPGEN is invalid.

User Response: Reduce the configuration requirements to within the system-assigned limits. For example, if the overflow is in the station area, reduce one of the following:

- The number of stations
- The number of segment headers per station
- The size of the return address stacks
- The size of the APCALL return address stacks

User segments, although part of the total storage requirements, are not part of the data areas described above.

BDK566I SEGMENT (segment) ON STATION (station) GT 65535

Explanation: The user segment specified is greater than the allowable maximum.

System Action: The configuration procedure ends before completing.

User Response: Correct the error and reassemble.

BDK567I OPERAND (keyword) IMPROPERLY SPECIFIED

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BKD568I CODE (scancode) IMPROPERLY SPECIFIED; (reason)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK569I WARNING: INITIALIZATION IGNORED (reason)

Explanation: Self-explanatory.

System Action: The configuration procedure completes, but the result is unpredictable.

User Response: Correct the error and reassemble.

BDK570I CPGEN COMPLETED SUCCESSFULLY

Explanation: Self-explanatory.

System Action: The configuration procedure completes.

BDK571I REQUIRED CPGEN MACRO (macro) OMITTED

Explanation: Self-explanatory

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK572I OPTIONAL MODULE (module) INVALID

Explanation: You specified an invalid optional module ID that is invalid for the 4700 system.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK575I TOO MANY TYPEMATIC AND/OR TRANSIENT SHIFT KEYS
SPECIFIED; MAXIMUM IS 16**

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK576I ONLY 7 CHARACTERS MAY BE SPECIFIED FOR A KEY

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK577I ERROR IN OPERAND NUMBER (number)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK578I POSITION NUMBER GREATER THAN SPECIFIED
CHARACTER SET**

Explanation: The position number specified in the OUTRTBL macro is greater than the maximum number of positions in the character set specified.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK579I TOO MANY OPERANDS SPECIFIED. MAXIMUM ALLOWED
IS (number)**

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Recode the macro in the controller configuration with fewer operands and reassemble.

BDK581I (macro) LABEL OMITTED

Explanation: The required label was not specified.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK582I (macro) LABEL GREATER THAN 8 CHARACTERS

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK583I DUPLICATE (macro) LABEL ASSIGNED

Explanation: The same label has appeared twice.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK584I CONFLICTING SUB-ADDRESS SPECIFICATION FOR DEVICE
(device)**

Explanation: The slot addressed by the device specified is either nonsharable or is already owned by a different slot-sharing type of device.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK585I DEFADDR DEVICE LABEL REFERENCE (label) DOES NOT
HAVE A MATCHING DEVXXXX MACRO**

Explanation: Self-explanatory.

System Action: The configuration process ends.

User Response: Correct the error and reassemble.

BDK586I INVALID (what) SPECIFIED

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Recode the macro and reassemble.

**BDK587I DEFADDR DEVICE LABEL REFERENCE (label) IS FOR A
VARIABLE ADDRESS DEVICE AND REQUIRES SUBADDRESS
SPECIFICATIONS**

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK588I DEFADDR MACRO (label) REFERS TO AN UNDEFINED
LOOPS MACRO (label)**

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK589I DEFADDR MACRO (label) AND DEFADDR MACRO (label)
HAVE A CONFLICTING TLA/MOD ASSIGNMENT ON LOOP
(number) FOR SLOT (slot number)**

Explanation: This slot number has already been assigned to a terminal or terminal group.

System Action: The configuration process ends.

User Response: Correct the error and reassemble.

**BDK590I DUPLICATE SUBADDRESS SPECIFICATION FOR DEFADDR
MACRO (label)**

Explanation: Self-explanatory.

System Action: The configuration process ends.

User Response: Correct the error and reassemble.

**BDK591I SUBADDRESS SPECIFIED FOR (labeldev) WHICH IS NOT A
SLOT-SHARING DEVICE**

Explanation: Self-explanatory.

System Action: The configuration process ends.

User Response: Correct the error and reassemble.

**BDK592I SUBADDRESS SPECIFICATION FOR (labeldev) IS NOT AN
EVEN/ODD PAIR**

Explanation: The subaddress for the 3608 display should be an even address; the subaddress for the printer should be an odd address.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK593I INVALID MODULUS SPECIFICATION FOR DEFADDR MACRO
(label) AND DEVICE REFERENCE (labeldev)**

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK594I DEFADDR MACRO (label) REFERS TO A NONSHARED
DEVICE (label)**

Explanation: An attempt has been made to assign a nonsharable device to a terminal address which has been designated as sharable.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK595I REQUIRED POSITIONAL OPERAND OMITTED

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK597I WARNING: SEGSTOR MACRO FOR CLASS (id) OMITTED

Explanation: Self-explanatory.

System Action: The configuration procedure completes.

User Response: Correct the error and reassemble.

**BDK601I WARNING: CONFLICTING OPTIONS (option1) AND (option2)
FOR (keyword) PARAMETER**

Explanation: Self-explanatory.

System Action: The configuration procedure completes, but system operations might not be as expected.

User Response: Review the macro definition and correct the code to eliminate conflicts; then reassemble.

**BDK606I WARNING: UNDEFINED STATION ID (id) DELETED FROM
PRIORITY DISPATCH TABLE (table)**

Explanation: The specified station id is not defined in this configuration.

System Action: The configuration procedure completes.

User Response: Correct the error and reassemble.

BDK607I LOW TO HIGH RANGE NOT CORRECTLY SPECIFIED

Explanation: You can specify a value range only as low-high.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK609I PRIORITY DISPATCHING TABLE EXCEEDS MAXIMUM LIMIT OF 256 ENTRIES

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK610I WARNING: TRANSLATE TABLE NOT SPECIFIED FOR MAGNETIC STRIPE ENCODER, DEFAULT ASSUMED

Explanation: The 4704 or 3604 terminal is defined with an encoder, but only an input magnetic stripe table is specified.

System Action: The configuration procedure completes, but the read and encode operations might not be compatible.

User Response: Default both translation tables for MSR and MSE, or specify both tables. If default tables are acceptable, you can specify them by name (BUAIMST for MSR and BUAOMST for MSE) to eliminate the diagnostic message.

BDK612I WARNING: STARTUP SPECIFIED ON PREVIOUS PRIORITY DISPATCH TABLE; IGNORED

Explanation: Only one priority table can be marked as active/nonactive at startup time.

System Action: The configuration procedure completes.

User Response: Correct the error and reassemble.

BDK613I WARNING: DEFINED STATIONS (id) NOT IN PRIORITY DISPATCH TABLE (id); ADDED

Explanation: All stations in the configuration must appear at least once in all priority dispatch tables.

System Action: The id is added to the table and the configuration procedure completes.

User Response: Correct the error and reassemble.

BDK614I PROCESS ERROR IN MACRO (macro); (reason)

Explanation: This message indicates that the macro named has detected an error of unexpected condition on an internal call from another controller configuration macro.

System Action: The configuration procedure ends.

User Response: Refer the problem to your service facility for correction and to get a possible bypass.

BDK616I WARNING: SSS TRANSMISSION REQUIRES EIGHT CHARACTER ID, SPECIFIED ID (id) PADDED WITH ZEROS

Explanation: Self-explanatory.

System Action: The configuration procedure completes after the name is padded with zeros.

User Response: None is required if the padded name is acceptable. If it is not acceptable, recode the name of the controller configuration with an eight-character name.

BDK623I SPECIFICATION OF (macro name) MACRO (operand name) OPERAND CONFLICTS WITH (macro name) MACRO (operand name) OPERAND.

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK624I DUPLICATE DEVICE COMPONENT DEFINITION ON MACRO
(macro name) and MACRO (macro name)**

Explanation: The same component of a device was assigned more than once.

System Action: The configuration procedure ends.

User Response: Check the A/B station indicator assignment on the LDA macro or in the TERM keyword of the STATION macro (or both places). Generally, you should have as many LDA macros as there are components being assigned to a station. Correct the error and reassemble.

**BDK625I INTERNAL ARRAY OVERFLOW AT (label) IN THE (macro name)
MACRO**

Explanation: A configuration array has been exceeded.

System Action: The configuration ends.

User Response: Inform your service facility, or remove the macro named in the error message.

| Alternate Line Adapter Messages

Note: Text in any of the following messages that refers to “SNAP” (formerly an RPQ program) now means “Alternate Line Adapter Feature”.

**BDK633 INVALID STATION IDENTIFIER SPECIFIED FOR (keyword)
(station id)**

Explanation: The station id specified is invalid. It contains non-numeric characters or is not in the range of 2-60.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK634 INITIAL STATE PARAMETER INVALID FOR (keyword) (state)

Explanation: Self-explanatory

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK635 (networkid) IS AN INVALID NETWORK IDENTIFIER (keyword)

Explanation: X'0000' and X'FFFF' are invalid network identifiers.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK637 (macro), (keyword) OPERAND CANNOT BE GREATER THAN ONE ADDRESS (label)

Explanation: Self-explanatory.

System Action: Second address is ignored. The configuration procedure completes.

User Response: None required. (The CPGEN is operational.)

BDK638 MULTIPLE DEFINITION OF (macro) LABEL (label)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK639 (macro) MACRO DEFINED BUT UNREFERENCED (label)

Explanation: Self-explanatory.

System Action: The configuration procedure completes.

User Response: None required. (The CPGEN is operational.)

BDK640 MAXIMUM (macro) MACRO SPECIFICATION (maximum number)

Explanation: The macro indicated has been specified greater than the allowable maximum number.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK642 NETID PREVIOUSLY DEFINED IN (macro) MACRO (netid)

Explanation: The netid associated with this macro has been defined previously in the macro specified.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK643 (keyword) OPERAND OF ALALINE SPECIFIED NETID FOR DEVICE NOT DEFINED (netid)

Explanation: Self-explanatory.

System Action: The configuration procedure completes.

User Response: None required. (The CPGEN is operational.)

BDK644 (keyword) OPERAND OF ALALINE SPECIFIED NETID ASSOCIATED WITH LINE

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK645 CONTROL UNIT AND TERMINAL SPECIFIED AS ACTIVE (control unit netid) (terminal netid)

Explanation: Either the control unit or the terminals, but not both, can be specified as active.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK647 (keyword) OPERAND SPECIFIED AS GREATER THAN (maximum number) HEXADECIMAL DIGITS

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK648 (macro), (keyword) OPERAND GREATER THAN (maximum size)
OR EQUAL TO ZERO (value)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK649 (macro) MACRO INSTRUCTION REFERENCES NONEXISTENT
(macro) LABEL (label)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK650 (keyword) OPERAND NOT SPECIFIED AS EVEN NUMBER OF
HEXADECIMAL DIGITS (value)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK651 (keyword) OPERAND NOT GREATER THAN (keyword2)
OPERAND (value1) (value2)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK652 DUPLICATE SEL ADDRESS SPECIFIED, ALACU LABEL (label)

Explanation: A Select Address specified for a terminal has already been specified for another terminal on the same control unit.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK653 INVALID AMS DISPLACEMENT SPECIFIED FOR STATION
(station id)**

Explanation: The AMS Displacement specification is either not a multiple of two or allows less than 8 bytes in the segment.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

**BDK654 INVALID AMS SEGMENT SPECIFIED FOR STATION (station
id)**

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK655 DUPLICATE DEFINED PORT VALUE (port number)

Explanation: The port number in the ALALINE macro has been previously defined.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK657 ALA NOT FUNCTIONAL, ALA LINE NOT SPECIFIED

Explanation: ALA was specified in the CPGEN without the specification of a line. An operand of a macro specified ALA, or an ALA macro instruction was specified.

System Action: The configuration procedure completes.

User Response: None required. (The CPGEN, with the exception of ALA, is operational.)

BDK659 (macro), (keyword) OPERAND IS INVALID (operand)

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK661 ALATERM (macro) CANNOT BE SPECIFIED IF (macro) PUNID IS NOT SPECIFIED

Explanation: Either the ALACU macro must specify a PUNID, or you must make sure that an ALATERM macro does not refer to the ALACU macro.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK663 ALACU (macro) SPECIFIED INVALID LINELN.

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK664 SNAP ALABUFF (macro) SPECIFIED CNL LT 20

Explanation: The minimum length allowed is 20 bytes.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK665 ALACU (macro) POLL CAN NOT BE SPECIFIED AS X'00' OR X'FF'

Explanation: Self-explanatory.

System Action: The configuration procedure ends.

User Response: Correct the error and reassemble.

BDK667 INVALID AMS SEGMENT INDEX SPECIFIED FOR STATION *stationid* IS FOUND

Explanation: The AMS segment index specification is incorrect.

System Action: The configuration procedure ends.

User Response: Correct the *ssid* specification in the AMS= parameter of the STATION configuration macro.

Appendix B. LCF Messages

This appendix contains the LCF messages in numeric sequence, along with explanations and recommended actions.

Note that most LCF messages cause the cursor to be displayed along with the message. When this occurs, press the Enter key to remove the message and continue processing.

07 INVALID REQUEST - REENTER

Explanation: You entered an LCF request that is invalid; LCF waits for you to enter a valid request.

User Response: Enter a valid LCF request or correct the spelling.

14 LENGTH EXCEEDS 80 - REENTER

Explanation: You entered more characters than a panel can contain. LCF works only with 80-character card images.

User Response: Enter a correct 80-character panel.

15 STMT NOT SUPPORTED

Explanation: You entered a CPGEN statement not processed by LCF.

User Response: See the LCF limitations in Chapter 4 and enter only CPGEN statements that are valid for LCF.

16 INVALID STMT

Explanation: The CPGEN statement is not valid, does not conform to that statement's syntax requirements.

User Response: Be sure that the statement matches the input from which you are working. If they match, your input is wrong; consult the programmer. If they do not match, correct the statement.

17 INVALID KEYWORD

Explanation: The last-displayed keyword is not valid.

User Response: Correct the keyword, and reenter the statement.

18 REQUIRED KEYWORD MISSING

Explanation: You omitted a keyword that is required for this CPGEN statement.

User Response: If you do not have that keyword, notify the programmer. Enter the statement again, with the required keyword.

19 DUPLICATE LABEL

Explanation: You have used this statement label elsewhere in the CPGEN file.

User Response: Use another statement label, and be sure that all your statement references are correct.

20 REQUIRED LABEL MISSING

Explanation: You omitted a required label on a CPGEN statement. This CPGEN statement must have a label.

User Response: Reenter the statement with a label.

21 INVALID LABEL

Explanation: You entered a statement label that is not valid; perhaps you used a character that is wrong, or a label that is too long.

User Response: Correct the label, and reenter the statement.

22 INVALID PARAMETER

Explanation: The last-displayed parameter is not valid.

User Response: Reenter the statement with the correct parameter.

23 MISSING DELIMITER

Explanation: You omitted a required delimiter from the CPGEN statement. Perhaps you have omitted a closing parenthesis.

User Response: Reenter the statement with all delimiters in place.

24 INVALID DELIMITER

Explanation: You entered a delimiter that is not valid in the statement.

User Response: Reenter the statement with correct delimiters.

25 KEYWORD VALUE ERROR

Explanation: You entered a parameter value outside the valid range. For example, perhaps you entered a 3 where the valid parameters are only 1 and 2.

User Response: Reenter the statement with the correct value.

27 STARTGEN STMT MISSING

Explanation: You omitted the STARTGEN statement from the file.

User Response: Use the LCF editor to enter a STARTGEN statement. Then, try again to create the operating diskette.

28 nnnnnnnn STMT - ONLY ONE ALLOWED

Explanation: The cited statement type has been entered more than once in this file. Only one is permissible.

User Response: Remove the extra statement using the editor, and try again to create an operating diskette.

29 JOB TERMINATED - ERRORS FOUND

Explanation: LCF cannot create an operating diskette because of the errors encountered. LCF ends processing.

User Response: Correct all errors. Use this procedure again to create an operating diskette.

40 DSKT ERROR

Explanation: A diskette read or write error occurred when LCF attempted to use a diskette.

User Response: Ensure that the correct diskette is inserted, and rerun the LCF job. If the error persists, inform your service representative.

41 LCF LOGIC ERROR

Explanation: The problem is internal to LCF.

User Response: Inform your service representative.

42 ERROR IN STMT - CHECK FINAL KEYWORD SHOWN

Explanation: You have made an error in the statement. The error is in the last keyword operand shown on the screen.

User Response: Correct the error using the LCF editor, and reprocess the file.

44 SYSLCF ALLOC FAILED

Explanation: LCF could not allocate the configuration data set (SYSLCF) on the LCF pre-operating diskette. The pre-operating diskette has been damaged, or it does not have enough space.

User Response: Use a new pre-operating diskette. (Each installation should have several backup pre-operating diskettes.)

45 NR. OF TRACKS TOO LARGE

Explanation: You attempted to allocate too many tracks for your SYSAP data set. This data set will contain your application programs (APs).

User Response: Consult the programmer for the correct number. The programmer can find the size of each program in the SYSAP directory of the AP diskette. Chapter 4 shows the format of the directory.

46 NR. OF APS TOO LARGE

Explanation: You attempted to allocate space for too many programs in your SYSAP data set.

User Response: Consult the programmer for the correct number. The programmer can find the size of each program in the SYSAP directory of the AP diskette. Chapter 4 shows the format of the directory.

47 SYSAP ALLOC FAILED

Explanation: LCF could not find enough space on the pre-operating diskette to allocate the SYSAP data set.

User Response: Run the COMPRESS operator function with the pre-operating diskette. The COMPRESS function is on the installation diskette. Then rerun LCF. If the problem persists, inform your service representative.

48 AP DSKT CONTAINS NO APS

Explanation: You inserted a diskette that contains no programs to be transferred.

User Response: Inform the person who supplied the application-program diskette.

49 MAX NR. OF APs EXCEEDED

Explanation: You attempted to add too many programs to a pre-operating diskette.

User Response:

1. Use the COPY DATASET operator function on the installation diskette to save the SYSAP data set on the pre-operating diskette (be sure you copy it to another diskette).
2. Use the LCF AP TRANSFER option to reallocate the SYSAP data set on the pre-operating diskette specifying a larger maximum number of allowed programs.
3. Use the LCF AP TRANSFER option to transfer the programs you saved in step 2 back to the pre-operating diskette.
4. Re-run the original job to transfer the new program to the pre-operating diskette.

50 PD SYSAP TOO SMALL

Explanation: You attempted to transfer a program that is too large for the current SYSAP data set on your operating diskette.

User Response: See message 49. Save the existing APs; reallocate SYSAP with a larger number of tracks; restore the existing APs; and, rerun the job.

51 SYNTAX ERROR IN STMT FILE

Explanation: The LCF Create Operating Diskette option has found a syntax error in the CPGEN statement file. The statement in error is displayed along with this error message.

User Response: Use the LCF editor to correct the error and then rerun the job.

52 CONT. STMT FORMAT ERROR

Explanation: You made an error in using a continuation panel.

User Response: Review the continuation rules in Chapter 4. Correct the file using the editor.

53 LOOP STMT NOT FOUND

Explanation: The LOOPS statement referred to by a DEFADDR statement does not exist.

User Response: Use the LCF editor to add the LOOPS statement, and rerun the job.

54 INVALID SUBADDRESS

Explanation: The subaddress on a DEFADDR statement was greater than 15, or a syntax error occurred on a DEFADDR statement (see Chapter 3 for DEFADDR syntax restrictions).

User Response: Correct the subaddress using the editor; reprocess the file.

55 GLOBAL SEGMENT: DO NOT SPECIFY ID

Explanation: You specified Segment 15 along with a specific station ID in an ENDGEN or SEGINIT statement. Segment 15 is always global and is not associated with any one station ID.

User Response: Use the editor to remove the ID parameter from the ENDGEN or SEGINIT statement.

56 GLOBAL SEGMENT NOT ALLOCATED BY STARTGEN

Explanation: You did not allocate a global segment.

User Response: Specify the S13/S15 parameter in the STARTGEN statement.

57 UNDEFINED STATION

Explanation: No STATION statement was found for the station ID specified by an ENDGEN or SEGINIT statement.

User Response: Use the editor to add the correct STATION statement, or correct the ENDGEN or SEGINIT statement.

58 SEGMENT 0 NOT SPECIFIED AS SHARED

Explanation: You did not specify that segment 0 is shared.

User Response: Use the editor to specify SHARED=YES in the STATION statement.

59 STORAGE CLASS CONFLICT

Explanation: A conflict was found between the storage class associated with the station ID or segment 13 and the SEGSTOR statements in the CPGEN statement file.

User Response: Verify that there is a SEGSTOR statement for the class associated with the station ID or the class specified for a non-global segment 13, and that the segment specified in the ENDGEN or SEGINIT is allocated by SEGSTOR.

60 SGMT NOT ALLOCATED BY ASSOCIATED SEGSTOR

Explanation: The ENDGEN or SEGINIT statement specified a SEGSTOR statement that did not allocate space for the specified segment.

User Response: Make necessary changes to the SEGSTOR or to the ENDGEN or SEGINIT statement.

61 DISP EXCEEDS SEGMENT LENGTH

Explanation: The displacement into the segment is larger than the segment.

User Response: Increase the segment's size or reduce the data or displacement.

62 DATA OVERFLOW OF SEGMENT

Explanation: The data has overflowed the segment.

User Response: Increase the segment's size or reduce the data or displacement.

63 ERROR IN DATA DESCRIPTION

Explanation: LCF allows only the following data descriptions:

```
ddd'txxx...x'  
tLnn'xxx...x'
```

ddd is the repetition factor (optional).
t is either X (for hex) or C (for character).
nn is the length of the constant (optional).
xxx...x is the data

User Response: Check the data description.

64 LCF ERROR - 0001

Explanation: An error within LCF was encountered.

User Response: Inform your service representative.

65 NO STMT FOUND WITH THIS LABEL

Explanation: This statement refers to a statement label that is not in the file.

User Response: Use the editor to correct this reference, and reprocess the file.

66 BUA PREFIX ON LABEL NOT ALLOWED

Explanation: You used a label (a symbolic name) on a CPGEN statement beginning with the letters BUA. BUAXxxx labels are reserved for system use.

User Response: Change the symbolic name.

70 MOUNT PD DSKT ON PRIMARY

User Response: Insert the pre-operating diskette on the primary drive for use in subsequent LCF processing.

71 MOUNT PD DSKT ON AUXILIARY

User Response: Insert the pre-operating diskette on the secondary drive for use in subsequent LCF processing.

72 MOUNT AP DSKT ON PRIMARY

User Response: Insert the application program diskette on the primary drive for subsequent LCF processing.

74 MOUNT BASIC EXCHANGE DSKT ON PRIMARY

User Response: Insert a basic exchange diskette on the primary drive. This diskette should contain one or more complete CPGEN files to be included on the installation diskette. (You can only transfer one CPGEN file.)

75 MOUNT BASIC EXCHANGE DSKT ON AUXILIARY

User Response: Insert a basic exchange diskette on the secondary drive. This diskette should contain one or more complete CPGEN files to be included on the installation diskette. (You can transfer only one CPGEN file.)

76 ENTER DATASET NAME

User Response: Enter the name of the data set that contains the CPGEN file that you want LCF to read from the basic exchange diskette.

77 DATASET NOT FOUND

Explanation: You tried to have LCF read a complete CPGEN file from a diskette. The data set you want read does not exist on the basic exchange diskette. Perhaps you inserted the wrong diskette or entered the wrong data set name.

User Response: Be sure that you insert the correct diskette that contains the data set you want LCF to read. Press Enter to continue.

78 AWAITING HOST CONTACT

Explanation: LCF is waiting for contact from the host system to begin transferring a CPGEN file from the host to LCF.

79 EXTRA BIND RECEIVED

Explanation: During an attempt to receive a CPGEN file from a host, you received an unexpected bind request.

User Response: Inform the person who supplied the host-transmit program.

80 EXTRA SDT RECEIVED

Explanation: During an attempt to receive a CPGEN file from a host, you received an unexpected SDT.

User Response: Inform the person who supplied the host-transmit program.

81 LCF IS IN SESSION WITH THE HOST

Explanation: LCF is receiving a CPGEN file from the host system.

82 UNEXPECTED DATA RECEIVED

Explanation: During an attempt to receive a CPGEN file from a host, you received data when it was not expected.

User Response: Inform the person responsible for the host-transmit program.

83 CHAIN ERROR

Explanation: During an attempt to receive a CPGEN file from a host, you received a middle-in-chain or last-in-chain record before a first-in-chain record.

User Response: Inform the person responsible for the transmit program.

84 PGM CHECK IN LCF BUCZQxxx nnnnnn

Explanation: An internal LCF error has occurred. BUCZQxxx indicates the actual LCF module in error.

User Response: Report the problem and the error code to your service representative.

85 KEYBOARD READ ERROR

Explanation: An error occurred while LCF was reading data from the keyboard.

User Response: Press reset and reenter the data. If the error continues, inform your service representative.

87 READ ERROR - REENTER

Explanation: An error occurred while LCF was reading your data.

User Response: Reenter the data.

88 COMPILE IN PROGRESS

Explanation: Informational message from LCF option 3; no response necessary.

90 COMPILE CANCELLED — OPERATOR REQUEST

Explanation: Informational message from LCF; no response necessary.

91 STATEMENT NUMBER nnn

Explanation: The statement shown was found to be in error during the first pass of LCF option 3.

User Response: Use the editor to correct the error, and rerun the job.

92 ERROR SMSDST=xxxx

Explanation: This message contains device status information and is displayed with other error messages when device-related errors occur.

User Response: Save the error code for your system programmer in case the action specified for message 090 and 091 does not correct the problem.

97 MOUNT LCF DSKT ON PRIMARY

User Response: Insert the 4700 LCF program diskette on the primary drive.

98 OPEN ERROR - SYSAP

Explanation: The SYSAP data set on the AP diskette or the pre-operating diskette could not be opened.

User Response: If AP diskette: Inform the supplier of the AP diskette. If pre-operating diskette: Ensure that you allocated the SYSAP data set with option 1 of the LCF AP TRANSFER function.

99 ENTER AP NAME

Explanation: You are transferring an application program from an input diskette. LCF wants you to enter the name of the program to be transferred.

User Response: Enter the application program name.

100 AP NOT FOUND

Explanation: LCF cannot find the named application program on the diskette.

User Response: Either insert the correct diskette or correct the application program name. Retry the operation.

101 AP EXISTS - ENTER R TO REPLACE

Explanation: You are transferring an application program to LCF, but that program already exists.

User Response: To replace the old program with the new program, enter the character R. If you do not want to replace the program, return to the main LCF menu.

102 AP TOO LARGE TO REPLACE

Explanation: You are attempting to replace an AP with a larger AP of the same name.

User Response: Inform the person who supplied the AP diskette.

104 DUPLICATE STATION IDS

Explanation: The CPGEN statement file contains at least two STATION statements with the same station ID.

User Response: Use the LCF editor to correct the STATION statement in error.

105 DUPLICATE LOOP IDS

Explanation: The CPGEN statement file contains at least two LOOPS statements with the same loop ID.

User Response: Use the LCF editor to correct the LOOPS statement in error.

107 INVALID DCA PORT

Explanation: You specified a port number greater than 15 in a DCAPORT statement.

User Response: Use the LCF editor to correct the DCAPORT statement.

108 DUPLICATE DCA PORTS

Explanation: You specified the same DCA port on at least one other DCAPORT statement.

User Response: Correct the DCAPORT statement in error.

109 DSKT WRITE ERROR

Explanation: An error occurred while LCF was writing to a diskette.

User Response: Retry the operation. If it continues to fail, inform your service representative.

111 DCAPORT NOT FOUND

Explanation: The DCAPORT macro referred to by the STATION macro's DA parameter does not exist.

User Response: Use the LCF editor to add the DCAPORT macro, and rerun the job.

112 DEFFADDR/LDA NOT FOUND

Explanation: The DEFFADDR or LDA macro referred to by the STATION macro's TERM parameter does not exist.

User Response: Use the LCF editor to add the missing macro, and rerun the job.

113 SYSPF ALLOCATION FAILED

Explanation: The pre-operating diskette lacks the space to allocate the permanent file.

User Response: Inform the person who supplied the CPGEN file. Possible actions are:

1. Reduce the permanent file size parameter on the FILES statement.
2. Run the compress function with the pre-operating diskette. (See message 128.)
3. Use a new pre-operating diskette.

114 SYSTF ALLOC FAILED

Explanation: The pre-operating diskette lacks the space for the temporary file.

User Response: Inform the person who supplied the CPGEN statement file.
Possible actions are:

1. Reduce the temporary file size parameter on the FILES statement.
2. Run the compress function with the pre-operating diskette. (See message 128.)
3. Use a new pre-operating diskette.

115 ERROR - REQUIRED COMF OMITTED

Explanation: You specified an INDXC parameter on a FILES statement, but omitted the required COMF statement.

User Response: Correct the FILES statement and rerun the job.

116 ERROR - MISSING TEMP FILE

Explanation: A FILES statement specified temporary file options, but omitted the TF parameter from the statement.

User Response: Correct the FILES statement and rerun the job.

117 ERROR - TOO MANY TEMP FILES

Explanation: A FILES statement exceeded the maximum number of temporary files.

User Response: Correct the FILES statement and rerun the job.

119 ERROR - TOO MANY SUB-FILE INDICES

Explanation: A FILES statement specified more than 60 subfile indexes.

User Response: Correct the FILES statement and rerun the job.

120 ERROR - BAD SUB-FILE NUMBER

Explanation: A subfile was incorrectly specified in a FILES statement.

User Response: Correct the FILES statement and rerun the job.

**121 WARNING - PF NOT ALLOWED WHEN DKT=2, DEFAULT OF
0 TAKEN**

Explanation: A FILES statement specified DKT=2 and PF. You cannot specify a permanent file on the secondary diskette.

User Response: Correct the FILES statement and rerun the job.

122 ERROR - REQUIRED EDAM OMITTED

Explanation: A FILES statement specified the DKT=2 and TF keywords, but not EDAM=Y.

User Response: Correct the FILES statement and rerun the job.

123 ERROR - INVALID COMF OPERAND

Explanation: You specified the COMF operand incorrectly.

User Response: Correct the FILES statement and continue.

124 WARNING - MABR EXCEEDED

Explanation: The maximum aggregate bit rate (MABR) is exceeded.

User Response: This is a warning message. It does not cause the job to end. Inform the person who generated the CPGEN statement file.

125 MOUNT AP DSKT ON AUXILIARY

User Response: Insert the diskette containing the application programs (APBs) into the secondary drive.

126 OPEN ERROR ON SYSSM

Explanation: LCF, in transferring the expanded system monitor to the pre-operating diskette, could not open data set SYSSM. The pre-operating diskette is probably in error.

User Response: Use one of your backup pre-operating diskettes.

127 READY RECEIVED FROM HOST

Explanation: A host system has established contact with the 4700.

User Response: If you are waiting to receive a CPGEN file from the host, enter option 5 on the main LCF panel to receive the CPGEN file. Otherwise, ignore this message.

**128 DELETE UNNEEDED OPTMODS (DEL TO DELETE, K TO
KEEP ALL OPTMODS)**

Explanation: This prompt allows you to delete unnecessary optional modules from the pre-operating diskette to free space for user data sets (programs and user-generated data sets), and to allow you to allocate larger temporary and permanent files. If you choose to delete unnecessary modules, make sure that you have made an adequate number of backup pre-operating diskettes; future CPGENs might require some of the modules you delete.

User Response: Reply DEL to delete the unneeded OPTMODS, or K to keep them.

129 LDA CONFLICT

Explanation: LCF detected a conflict in logical device address assignments.

User Response: Examine your STATION macro and its TERM operands for errors, and consult your programmer.

130 STMT FILE EMPTY -- EDIT TERMINATED

Explanation: You invoked the LCF editor (option 2 on main LCF menu), but there is no CPGEN file to edit.

User Response: Create a CPGEN file to edit; use option 1.

131 DSKT MOUNTED IS NOT A VALID PD

Explanation: You were requested to mount a valid pre-operating diskette; the diskette that you mounted is not a pre-operating diskette.

User Response: Mount a valid pre-operating diskette when you are next prompted.

132 REQUESTED SYSTEM MONITOR NOT ON PD

Explanation: You specified a system monitor in your CPGEN information, but that system monitor is deleted from the pre-operating diskette.

User Response: Use another pre-operating diskette.

133 DUPLICATE SEGSTOR CLASSES

Explanation: You have specified more than one SEGSTOR statement with the same class.

134 APCALL FROM LCF DSKT FAILED

Explanation: LCF could not read the LCF program diskette. Perhaps the LCF program diskette was removed from the diskette drive and another diskette inserted in its place.

User Response: Re-IPL the LCF program diskette. Make sure that you change diskettes only when prompted by LCF, and that you mount the correct diskette.

135 WARNING: AT LEAST ONE REQUESTED MOD MISSING FROM PD

Explanation: You previously deleted one or more needed optional modules from the pre-operating diskette.

User Response: Rerun LCF with a new pre-operating diskette.

136 WARNING: STATEMENT FILE TRUNCATED

Explanation: LCF cannot read the complete statement file from the LCF program diskette because of insufficient controller storage.

User Response: Specify optional modules 1B, 1D, 5E, and 68 during IPL of the installation diskette, or run the LCF job on a controller with more storage.

137 ERROR: INVALID PARM COMBINATION

Explanation: Two or more parameter values on the CPGEN statement violate some rule governing the dependencies between these parameters.

User Response: Consult the CPGEN statement description in Chapter 3 and determine the correct inter-parameter values. For example, the DEV4720 statement limits the PS parameter to 65 lines if the LPI parameter was specified as 5.

138 DSKT MOUNTED IS NOT LCF DSKT

Explanation: You did not insert a valid LCF program diskette.

User Response: Insert a valid LCF program diskette.

139 WRONG DSKT MOUNTED

Explanation: You inserted the wrong kind of diskette.

User Response: Insert the correct diskette type (pre-operating diskette or basic exchange diskette); LCF continues the job.

140 TOO MANY LABELS IN STMT FILE

Explanation: Your CPGEN statement file exceeds LCF's restriction on the number of statement labels.

User Response: Remove any unnecessary labels with the editor, and rerun the job.

141 KEYWORD DEPENDENCY ERROR

Explanation: Two or more keywords on the statement being processed have dependencies, and an error has occurred for this reason.

User Response: Consult the description for using the statement or macro and note the dependencies. Then correct the error, and rerun the job.

142 REQUIRED X25 STATEMENT MISSING

Explanation: You specified TYPE=X25 on the COMLINK statement, but did not include either an X25LCHN or X25CKT statement.

User Response: Correct the problem, and rerun the job.

144 MOUNT LIB DSKT ON PRIMARY

User Response: Mount a library diskette on the primary drive.

145 MOUNT LIB DSKT ON ALTERNATE

User Response: Mount a library diskette on the alternate (logical secondary) drive.

146 ENTER NAME FOR STATEMENT FILE

User Response: Enter the name (1-8 alphameric characters) by which you want the statement file saved on the library diskette.

147 DATASET EXISTS. REPLACE(Y/N)

Explanation: The name you selected for your statement file already exists for another data set on the library diskette.

User Response: Enter N to force LCF to ask you for another name. Enter Y if you want to replace the existing file of that name with your new file on the library diskette. Enter a null response to cause no action to take place.

148 ENTER STATEMENT FILE NAME

User Response: Enter the name of the data set you want to retrieve from the library diskette.

149 DATASET NOT FOUND

Explanation: The data set with the name you specified for retrieval is not on the library diskette.

User Response: Display the names of the data sets on the library diskette to determine whether the data set you want is on the diskette, or that you have correctly spelled the name of the data set you want.

150 ENTER DATASET NAME TO DELETE

User Response: Enter the name of the data set you want to delete from the library diskette.

151 SAVE CONFIG DS TO LIB DSKT (Y/N)

User Response: Enter Y to save the LCF option 3 output to the library diskette. LCF will then assign the name by appending the suffix *.LCF* to the CPGEN name specified in the STARTGEN statement.

152 X25 PARAMETER SPECIFIED BUT X25 TYPE NOT SPECIFIED

Explanation: Your COMLINK macro contained one or more keywords that apply only to X25, but you failed to code TYPE=X25.

User Response: Correct your macro specification, and rerun the job.

154 HORIZONTAL ADJACENT DOTS SPECIFIED

Explanation: Your CHARDEF macro specified a character that has horizontally adjacent dots, a condition that the printer cannot handle.

User Response: Correct your macro specification, and rerun the job.

184 2 MSTRTBL LABELS REQUIRED

Explanation: The DCAXxxx statement's MSTRTBL keyword requires both an input and an output MSTRTBL label.

User Response: Press the Enter key to continue.

185 I/O ERROR - REQUEST TERMINATED

Explanation: An unrecoverable I/O error occurred during LCF processing. The job ends. This message might follow message 92 (the device-status message).

User Response: Inform your service representative.

186 REQUIRED SUBADDRESS MISSING

Explanation: A 3616 or 4710, or a 3615 with the MG1586 option, was referred to by a DEFADDR macro that did not specify the necessary subaddress.

User Response: Correct the coding of the DEFADDR macro, and rerun the job.

| Appendix C. Alternate Line Attachment Feature Configuration Macro Descriptions

This section describes the configuration macros needed during the CPGEN process to define the Alternate Line Attachment Feature (RPQ 8V0132) to the 4700 system controller. The macros define those characteristics of the 4700 system that are unique to Alternate Line Attachment (ALA). You can also use this information to define the feature that supports communication with the 4730 Personal Banking Machine.

The syntax notation used in describing the macros is identical to the notation used in the *4700 Controller Programming Library* descriptions of assembler instructions and configuration macro instructions with one exception: operands that require specification of EBCDIC data can be coded directly either without any data type identification or in the form *C'operand'*.

The macros used by ALA are:

- ALABUFF — creates the ALA input buffer pool.
- ALALINE — specifies ALA line characteristics.
- ALACU — specifies the characteristics of the ALA physical units (PUs)
- ALATERM — specifies the characteristics of the ALA secondary logical units (SLUs).

The number and choice of configuration macros used depends on the number and type of lines and units to be attached. If a stand-alone terminal is to be attached, both an ALACU macro describing the PU type and an ALATERM macro describing the SLU is necessary. In other words, a stand-alone terminal is effectively a cluster controller with only one SLU.

The 4701 (like the 3694) can be equipped with a Fan-Out Communications Adapter that allows as many as four controllers or controller-level devices to be connected locally to an ALA line. To define this network, you must use one ALALINE macro and as many as four ALACU macros. The number of ALATERM macros depends on the number of SLUs per secondary PU.

Another choice is to equip the 4701 with a Local-Remote Communications Adapter, which allows attachment of either one locally-attached or as many as 254 remotely-attached controllers or controller-level devices.

To define this network, you must code one ALALINE macro, an ALACU macro for each attached physical unit (PU), and one ALATERM macro for each SLU within a PU.

| ALABUFF -- Create Buffers

ALABUFF creates a pool of input buffers that is associated with an ALA line. The parameters specify the number and size of the ALA input buffers in the pool. Use of ALABUFF is optional.

Name	Operation	Operand
label	ALABUFF	[CNL={length 100}] [,CNB={bufs 5}]

label

Required symbolic name (1-8 alphameric characters). The ALALINE macro uses ALABUFF's label to identify the buffer pool associated with the ALA line.

CNL

The *length* specification (a decimal number, 20 minimum) establishes the length of each input buffer. If you do **not** specify DBA=Y in the ALALINE macro, specify *length* on ALABUFF as a value equal to the maximum-size message that will be received. If you **do** specify DBA=Y in the ALALINE macro, specify *length* on ALABUFF as a value equal to the average-size message that will be received. The default value is 100.

Note: The macro processor automatically adds eleven bytes to your specification to provide space for the ALA headers.

CNB

The *bufs* specification (a decimal number greater than 0) establishes the total number of read buffers to be allocated to all ALA-attached devices. Specify a value at least large enough to provide buffers to accommodate the maximum-size input message if you code DBA=Y in the ALALINE macro. The default value is 5.

| ALALINE -- Define ALA Line

ALALINE specifies the physical characteristics of the ALA line. You must use ALALINE when generating an ALA configuration.

Name	Operation	Operand
label	ALALINE	NETID=networkid,LAM=SNAP [, ISTATUS={INACT ACT}] [, BUFFER=labelbuff] [, ACB={{ [PRT] [,DCE] }}] [, LNKTB={nnn 16}] [, CTG={nnn 1}] [, PSFC={nnnn 10}] [, RETRY={nnn 3}] [, SLOW={nnn 20}] [, DBA={Y N}] [, PORT={n 1}] [, WRAP={Y N}] [, DCL={nnnn 1200}]

label

Required symbolic name (1-8 alphanumeric characters). The ALACU macro uses this label to identify this line with the associated control units being specified.

NETID

The *networkid* is the two-character EBCDIC or four-character hexadecimal identification to be associated with this line. You specify this network identifier in the work station AMSNID field when requesting a start-line, stop-line, stop-soliciting-data, or resume-soliciting-data activity. The network identifier is valid only when used in an instruction that requires the referencing of an ALA line. It is invalid in any other instruction type.

LAM

Identifies the type of line discipline. Set the value to SNAP.

ISTATUS

Specifies the initial state of the line as active (ACT) or inactive (INACT) after system startup. If you specify the line as active, it starts during system initialization. The default value is ACT.

BUFFER

The *labelbuff* is a symbolic name (1-8 alphameric characters) of the ALABUFF macro (if coded) that specifies the characteristics of the buffer pool associated with this line. If you omit this operand, the line is associated with either the first ALABUFF macro coded or a default buffer pool consisting of one 120-byte buffer.

ACB

Specify PRT if you want a permanent request to send. Specify DCE if you want the data link communication to be done using NRZ encoding.

LNKTB

The *nnn* specification is a decimal number (within the range 2-500) that specifies the length of the buffer to be used for receiving SDLC link-test responses. This value determines the maximum length of the link-test frame that is transmitted. If you specify more than one line, the maximum length that you specify is used for all lines. The default value is 16 bytes.

CTG

The *nnn* specification is a decimal number (within the range 1-154) that specifies the number of 0.1-second increments that constitute a poll or write-time-out period. Specifying time-out periods of 0 (zero) or 1 forces a default 0.1-second time-out.

Note: The CTG, PSFC, and SLOW default values are the most suitable for communication over an ALA link.

PSFC

The *nnnn* specification is a decimal number (within the range 2-255) that specifies the number of 0.006-second increments for transmitting poll and select-sequence transmission. A maximum of 1.6 seconds is permissible. The default value is 10 (0.06 seconds).

Note: The CTG, PSFC, and SLOW default values are the most suitable for communication over an ALA link.

RETRY

This value is the number of time-out retries allowed for a poll or response time-out before a PU is declared to be down. The retry value is used for polling operations only. The default value is 3.

SLOW

The *nnn* specification establishes the minimum number of passes through the normal poll list before a PU in the slow-poll list is polled. Specifying 0 gives the same results as specifying 1. The maximum permissible value is 127. The default value is 20.

Note: The CTG, PSFC, and SLOW default values are the most suitable for communication over an ALA link.

The number of passes through the poll list for any network entity in slow-poll mode is unpredictable. The actual number is a factor of the slow value specified for the entity, the total number of entries in the poll list, and the number of entities concurrently failing. The “worst-case” value (longest time between polls) is the cumulative count of the slow values associated with all failing factors plus the number of entries in the poll list.

DBA

Y specifies that dynamic read buffer allocation is to take place if the input message exceeds the buffer size. You must not specify **DBA=Y** if you are using inbound segmenting. **N** specifies no dynamic read buffer allocation, and input messages larger than the buffer causes the buffer to overflow. **N** is the default value.

PORT

The *n* specification is the decimal number 1, which denotes the port assigned to this line. You must specify the **PORT** parameter on each **ALALINE** macro. The port has a 4700 physical-adapter address of 80.

WRAP

Specify **Y** if the line attaches to a “wrappable” modem. A wrappable modem performs a wrap test on both the adapter and the modem if a line failure occurs. If you specify **N**, the default, for a non-wrappable modem, only the adapter is wrap-tested.

DCL

Specify the line speed in bits-per-second as a decimal number. This value does not determine line speed, but is a value that **ALA** uses to provide the best line performance. The default value is 1200.

| ALACU -- Define ALA Control Unit

ALACU specifies the characteristics of an ALA control unit (CU).

Name	Operation	Operand
label	ALACU	LINE=linelabel ,NETID=networkid ,POLL=poladr1 [,LINELN={nnn 500}] [,RETRY={nnn 3}] [,ALAMAX={nn 1}] [,PUTYPE={1 2}] [,PUNID=(nid[,sta][,RSP])] [,ILVS={Y N}] [,LOG={Y N}]

label

Required symbolic name (1-8 alphameric characters) that one or more ALATERM macros can use to associate their SLUs with the CU defined by this ALACU macro.

LINE

The *linelabel* is a symbolic name (1-8 alphameric characters) of the ALALINE macro that defines the line associated with this CU.

NETID

The *networkid* is a two-character EBCDIC or four-character hexadecimal identifier associated with this CU. You specify this identifier in the workstation's AMSNID field when requesting an I/O operation or control operation.

POLL

The *poladr1* is a one-character EBCDIC or two-character hexadecimal value that specifies the address to be used in a polling sequence.

Note: For more efficient use of storage, specify CU addresses in a contiguous range. For example, specifying X'01' and X'02' generates a 16-byte address table. Specifying X'01' and X'FE' generates a 1024-byte address table.

LINELN

The *nnn* is a decimal number (in the range 1-500) that specifies the maximum number of bytes that can be written from the PLU's segment. This parameter is not used for the LU-LU session flow if you specify segmenting and are using the message-routing mode. The default value is 500.

RETRY

The *nnn* specifies the number of times that a failing operation is to be retried. The system updates the associated statistical counter with each occurrence. You can specify any value in the range 0-255, but no retry attempt occurs if you specify zero. The default value is 3. Note that this retry value is used for write operations and inbound BCC errors.

ALAMAX

The *nn* is a decimal number that defines the maximum number of write operations that can be outstanding for a stand-alone terminal at any one time. If a station issues a write request that exceeds this value, the station goes into the "defer state" until a previously-requested write operation completes. The default value is 1.

PUTYPE

Set this value to either 1 or 2 to indicate the physical unit (PU) type. The default is 2 for PU-Type 2 (FID3).

PUNID

The *nid* is a two-character EBCDIC or four-character hexadecimal identifier associated with the physical unit (PU) in this control unit. It has no default value. You specify this identifier in the work station AMSNID field for the subsystem control program (SSCP) or application program when reading or writing NS commands on the SSCP-PU session. Each network entity must have a unique network ID. The PUNID should therefore not be the same as the NETID.

The *sta* specification is a decimal number (in the range 2-31) that specifies the designated station for this PU. If you do not specify this, you must issue the LCNTRL ASN instruction with the *nid* value in AMSNID.

Specify RSP when you want the response data received from the ACTPU command passed to the SSCP or application program.

Note: Do not issue the LCNTRL VONL (vary-online) or LCNTRL VOFF (vary-offline) instructions to units identified by a PUNID. Additional information is in *Controller Programming Library — Volume 3*.

ILVS

Specify **Y** if segments for two or more SLUs associated with this PU can be interleaved. **N** (interleaving not allowed) specifies that, if one SLU associated with this PU is receiving a segmented message, messages to a second SLU associated with this PU will not be sent until the entire segmented message is sent to the first SLU. The default value is **N**.

LOG

If you want a log message written when an already-active control unit reports loss-of-contact, specify **Y**. Specify **N**, the default, if no message is to be written.

| ALATERM -- Define ALA SLU

ALATERM specifies the characteristics of an ALA secondary logical unit (SLU). You must code this macro if you want message routing to occur.

Name	Operation	Operand
------	-----------	---------

Name: Operation: Operand:

label	ALATERM	CU=labelcu ,NETID=networkid ,SEL=seladr1 [,STA=(nn [, {ACT INACT}])] [,ALAMAX={nn 1}] [,RSP={Y N}] [,PACING=nnn] [SEGMENT=nnn]
-------	---------	--

label

Optional label. Any symbolic name (1-8 alphameric characters) that has not already been used for the generation process.

CU

The *labelcu* specification is a symbolic name (1-8 alphameric characters) of the ALACU macro that defines the PU associated with this SLU.

NETID

The *networkid* is a two-character EBCDIC (coded C'xx') or a four-character hexadecimal (coded X'xxxx') identifier associated with the SLU. You specify this identifier in the PLU's AMSNID field when requesting an I/O operation.

SEL

The *seladr1* is a two-character hexadecimal value (coded X'xx') that specifies the address of this SLU. If you code PUTYPE=1 in the ALACU macro associated with this ALATERM macro, you must code a value in the range 0-63. Otherwise, a value in the range 1-255 is valid.

Note: The *seladr1* value is transmitted as the DAF in the TH of the outgoing message.

STA

Specifies whether the SLU is to be associated with a work station owned by or assigned to the free pool and, if owned, whether the initial state should be active (ACT) or inactive (INACT), the default.

The value for *nn* is a decimal number (in the range 2-31) that specifies the 4700 PLU station that is to own this SLU. If you specify this parameter, you need not issue an LCNTRL ASN instruction to indicate an acquire function in order to establish ownership. If you do not specify this parameter, the SLU goes into the free pool.

The choice between **ACT** or **INACT** specifies the initial state of this SLU as active (**ACT**) or inactive (**INACT**) after system startup if ownership has been established. If you specify the SLU as active, you establish message-routing mode, and initiate polling to the PU. If the PU responds, an optional activate-physical transmission goes to the PU, and an activate-logical transmission goes to the SLU. No **LCNTRL vary-online** instruction is necessary if you specify **ACT**.

Note: If the operational system is not to contain the optional control instructions, you must specify an owning station with this operand, and an initial state of **ACT**.

ALAMAX

The *nn* is a decimal number that defines the maximum number of write operations that can be outstanding for this SLU at any one time. If the station issues a write request, and this count or the count specified in the **ALAMAX** parameter of the **ALACU** macro is exceeded, the station goes into the “defer state” until an outstanding write operation completes. The default value is 1.

RSP

You specify **Y** if the response data received on the **ACTLU** command should be passed to the **SSCP-AP**. The default value is **N**.

PACING

The *nnn* is a decimal number in the range 1-63 that specifies the number of data messages that can be sent to the SLU on the LU-LU session before a pacing response must be received. The first message written requests the response. For example, if this value is 4, the first message requests the response, and no more than four messages could go out before a pacing response comes in. If you do not specify this parameter, no pacing takes place.

SEGMENT

The *nnn* specification denotes the smallest segmented message size that the SLU accepts. Specify a decimal number in the range 20-500. The recommended minimum message size is 128. If you do not specify a value, no message segmenting takes place.

| ALA Changes to 4700 CPGEN Macro Instructions

| *STARTGEN Macro*

The STARTGEN macro has one operand added:

```
... [ ,LCNTRL={Y|N} ]
```

LCNTRL

Specifies whether the optional control instructions, should be included in the system. **Y** indicates that the optional instruction should be included, and **N** indicates that it should not. The optional instruction support requires 2,000 bytes of storage. The default value is **N**.

| **STATION Macro**

The station macro has three operands added:

```
... [ ,ALA={Y|N} ]  
     [ ,ALAMAX={nn|1} ]  
     [ ,AMS=(seg,disp) ]
```

ALA

Specifies whether a station should be generated to communicate with ALA-ALA devices. **Y** indicates that the station uses ALA-ALA devices, while **N** indicates that it does not. If you specify **Y**, you must also specify the **AMS** operand. **N** is the default value.

ALAMAX

The *nn* is a decimal number greater than zero that defines the maximum number of write operations to ALA-ALA devices that can be outstanding for the station at any one time. If a station issues a write request while this count is exceeded, the station goes into the “defer state” until one of the outstanding write operations completes. You must specify **ALA=Y** if you specify **ALAMAX**. The default value is 1.

AMS

Identifies the segment and location within the segment of the ALA-ALA machine segment (**AMS**) for this 4700 work station. If you specify **ALA=Y**, you must also use this parameter.

- | | |
|-------------|---|
| seg | Identifies the segment that is to contain the AMS . Any segment except segment 14 is valid. The segment number, in hexadecimal, is available to the application program in the SMSAMS1 field. |
| ssid | Defines the segment space ID to which the defined AMS belongs. If you specify zero (the default), the defined AMS belongs to the current load segment space ID and is 10 bytes long. Other permissible <i>ssid</i> values in the range 1-14 specify the segment space ID to which the AMS belongs, and create a 16-byte AMS . |

Index

A

A command 4-13, 4-17
A operand (COMLINK) 3-9
A operand (LDA) 3-78
A operand (STATION) 3-108
AA operand (COMLINK) 3-10
AC operand (COMLINK) 3-10
ACB operand (ALALINE) C-4
ACB operand (COMLINK) 3-8
accent operand (DKATBL) 3-62
add configuration panel 4-37
adding programs 4-25
ADDR operand (COMLINK) 3-12
ADVANCE operand (DEV4720) 3-61
ALA operand (STATION) C-11
ALABUFF C-2
ALACU C-6
ALALINE C-3
ALAMAX operand (ALACU) C-7
ALAMAX operand (ALATERM) C-9
ALAMAX operand (STATION) C-11
ALATERM C-8
ALITE operand (DEV4704) 3-54
ALLOC operand (FILES) 3-71
allocating space 4-25, 4-26
ALT operand (DCATRTBL) 3-15
ALTLED operand (FILES) 3-72
ALTTBL operand (TRTLIST) 3-118
AMS operand (STATION) C-11
APBNM operand (STATION) 3-106
APL operand (DCA3278) 3-19
APL suboperand (DCA3279 FEATURE) 3-22
APLIST macro 3-3
APMEDIA macro 3-4
apname operand (APLIST) 3-3
APSTACK operand (STATION) 3-112
assembler conventions 4-8
assembler language conventions 4-2
asterisk (*) 4-11
ASYNLDA operand (STATION) 3-111
AUTO suboperand (DEV3616 NL) 3-49
AUTO suboperand (DEV4720 NL) 3-60
AVGML operand (DEV3608) 3-36

B

B command 4-14
B operand (COMLINK) 3-9
B operand (LDA) 3-78
B operand (STATION) 3-108
basic exchange diskette 4-3
basic exchange diskette, CPGEN files on 4-34
begin suboperand (DEV3616 CFOLD) 3-49
BIND keyword 3-110
BIND operand (COMLINK) 3-11
BPS operand (LOOPS) 3-80
BSC operand (COMLINK) 3-12
BSC operand (STARTGEN) 3-104
BSC operand (STATION) 3-111
BSCOPT operand (COMLINK) 3-12
BUF operand (FILES) 3-70
BUFFER operand (ALALINE) C-4
buffers operand (FILES) 3-71

C

C command 4-13, 4-15
CANCEL operand (TRTBHDR) 3-116
cancelling the operating diskette 4-23
cards, comment 4-9
cards, continuing 4-11
CASE operand (DCATRTBL) 3-16
CASE operand (INTRTBL) 3-76
CFOLD operand (DEV3611) 3-40
CFOLD operand (DEV3612) 3-43
CFOLD operand (DEV3616) 3-49
CHAIN operand (DCA3262) 3-17
char operand (INTRTBL) 3-74
char operand (MSTRTBL) 3-81
CHARDEF macro 3-5
CHARDEF operand (OUTBHDR) 3-87
CHARSET operand (DEV4704) 3-53
CHARSET operand (OUTRTBL) 3-88
CHARSET operand (OUTSPEC) 3-91
C'c' operand (PINTBL) 3-93
CLASS operand (SEGSTOR) 3-100
CNB operand (ALABUFF) C-2
CNB operand (COMLINK) 3-8
CNL operand (ALABUFF) C-2
CNL operand (COMLINK) 3-7
CNM operand (STARTGEN) 3-104
code operand (INTRTBL) 3-73
COMF operand (FILES) 3-69
COMLINK macro 3-7
commands, editing 4-13
comment cards 4-9
comp operand (EXTCTR) 3-66
component operand (LDA) 3-77
composite operand (DKATBL) 3-62
configuration macro instructions 1-1
configuration macros
 STARTGEN 3-101
CONSHR operand (DEV3610) 3-38
CONSHR operand (DEV3612) 3-43
CONSHR operand (DEV3615) 3-46
CONSHR operand (DEV3616) 3-49
continuation cards 4-11
continuation panel 4-37
control program generation 1-1, 4-1
CONV operand (COMLINK) 3-13
conventions, assembler 4-8
conventions, assembler language 4-2
correcting a statement 4-15
correcting errors 4-10
CPGEN 1-1, 4-1
CPGEN file 4-2
CPGEN file, create 4-7
CPGEN file, receiving panel 4-40
CPGEN files, receiving 4-33
CPGEN macros
 macro operands
 CPU (STATION) 3-110
CPGEN size 4-45
CPGID operand (OUTSPEC) 3-91
CPI operand (DEV3616) 3-50
CPI operand (DEV4710) 3-57
CPI operand (DEV4720) 3-60
CPU operand (STATION) 3-110

CPYAP operand (FILES) 3-70
create file 4-7
CTG operand (ALALINE) C-4
CTG operand (COMLINK) 3-9
CTLR operand (STARTGEN) 3-101
CU operand (ALATERM) C-8
CURSOR operand (DCA3278) 3-18
CURSOR operand (DCA3279) 3-21
CURSOR operand (DCA4704) 3-26
CURSOR operand (DEV3604) 3-33
CURSOR operand (DEV4704) 3-54

D

DA operand (STATION) 3-109
DASD operand (COMLINK) 3-12
data operand (ENDGEN) 3-65
data operand (SEGINIT) 3-99
data set name, CPGEN 4-34
datasets operand (FILES) 3-70
DATE operand (STARTGEN) 3-101
DBA operand (ALALINE) C-5
DC operand (COMLINK) 3-10
DCA suboperand (STATION DSM) 3-112
DCAPORT macro 1-9, 3-14
DCATRTBL macro 1-11, 3-15
DCA3262 macro 3-17
DCA3278 macro 3-18
DCA3279 macro 3-20
DCA3287 macro 3-23
DCA4704 macro 3-25
DCA5210 macro 3-24
DCE operand (COMLINK) 3-8
DCL operand (ALALINE) C-5
DCL operand (COMLINK) 3-7
DEBLSE operand (DEV3610) 3-39
DEBLSE operand (DEV3612) 3-43
DEF operand (COMLINK) 3-12
DEF operand (OUTRTBL) 3-88
DEF suboperand (STATION BSC) 3-111
DEFADDR macro 3-28
defaddr operand (LDA) 3-77
DEFCHAR operand (DEV3616) 3-50
DEFCHAR operand (DEV4710) 3-57
DEFCHAR operand (DEV4720) 3-61
DEFQUAL operand (DEV4720) 3-61
DEFSBF macro 1-11, 3-30
DEFSPA macro 1-10, 3-31
DEFTRT operand (DEV3616) 3-50
DEFTRT operand (DEV4710) 3-57
DEFTRT operand (DEV4720) 3-61
DEL command 4-14, 4-17
deleting a statement 4-17
DELSET operand (STATION) 3-109
DEL1 operand (COMLINK) 3-10
DEL2 operand (COMLINK) 3-10
DEVICE operand (CHARDEF) 3-6
DEVICE operand (DCATRTBL) 3-15
device operand (LDA) 3-77
DEVICE operand (OUTRTBL) 3-88
DEVICE operand (OUTSPEC) 3-91
DEV3604 macro 3-32
DEV3606 macro 3-35
DEV3608 macro 3-36
DEV3610 macro 3-38
DEV3611 macro 3-40
DEV3612 macro 3-42
DEV3614 macro 3-45

DEV3615 macro 3-46
DEV3616 macro 3-48
DEV3624 macro 3-52
DEV4704 macro 3-53
DEV4710 macro 3-56
DEV4720 macro 3-59
directory, diskette 4-44
diskette directory 4-44
diskette requirements 4-41, 4-44
diskette, installation 4-4
diskette, library 4-2, 4-3, 4-11
diskette, operating 4-22
diskette, pre-operating 4-4
diskette, receiving files on 4-34
diskette, transmitting 4-33
diskettes, mounting 4-24
disp operand (ENDGEN) 3-64
disp operand (SEGINIT) 3-98
DKATBL macro 1-11, 3-62
DKATBL operand (TRTBHDR) 3-116
DKT operand (FILES) 3-68
DPOOL operand (STATION) 3-113
DRBC operand (COMLINK) 3-12
DSK operand (FILES) 3-68
DSKOP operand (STARTGEN) 3-102
DSL operand (COMLINK) 3-8
DSM operand (STATION) 3-111
DTACC operand (STATION) 3-113
DUMP operand (STARTGEN) 3-103
DUP command 4-14

E

E command 4-14, 4-22
EAB suboperand (STATION DSM) 3-112
EDAM operand (FILES) 3-70
editing 4-3, 4-12
editing commands 4-13
editing panel 4-38
EID operand (INTRTBL) 3-76
EID suboperand (MSTRTBL func) 3-82
ENCRYPT operand (PINTBL) 3-93
END operand (PINTBL) 3-93
ENDGEN macro 3-64
ending the file 4-11
EOF operand (INTRTBL) 3-75
EOF suboperand (MSTRTBL func) 3-82
EOFA operand (INTRTBL) 3-75
EOFB operand (INTRTBL) 3-75
EOM operand (INTRTBL) 3-75
EOM operand (PINTBL) 3-94
EOM suboperand (MSTRTBL func) 3-82
EOMA operand (INTRTBL) 3-75
EOMB operand (INTRTBL) 3-75
EOMSET operand (DCA3278) 3-18
EOMSET operand (DCA3279) 3-20
EOMSET operand (DCA4704) 3-26
EOMSET operand (DEV3604) 3-33
EOMSET operand (DEV4704) 3-54
errors, correcting 4-10
ERTLS operand (DCA3278) 3-19
ERTLS operand (DCA3279) 3-21
ERTLS operand (DCA4704) 3-26
ERTLS operand (DEV3604) 3-34
ERTLS operand (DEV4704) 3-55
EXTCTR macro 3-66

F

F command 4-13, 4-16
FEATURE operand (DCA3278) 3-19
FEATURE operand (DCA3279) 3-22
FEATURE operand (DCA4704) 3-26
file name, CPGEN 4-34
file, CPGEN 4-2
FILES macro 3-68
files on basic exchange diskette 4-34
FILES statement 4-17
files, editing 4-12
files, terminating 4-11
FILL operand (PINTBL) 3-93
FLAG operand (COMLINK) 3-10
FONT operand (DEV4720) 3-60
FORMWD operand (DEV3608) 3-36
FT operand (DEV3610) 3-38
FT operand (DEV3612) 3-43
FT operand (DEV3615) 3-46
func operand (INTRTBL) 3-74
func operand (MSTRTBL) 3-82
func operand (OPTMOD) 3-85

G

generation, control program 1-1, 4-1
gpc operand (COMLINK) 3-12

H

host link 4-34
host network 4-33
host program requirements 4-41

I

ID operand (DEFSPA) 3-31
id operand (ENDGEN) 3-64
ID operand (EXTCTR) 3-67
ID operand (LOOPS) 3-80
ID operand (PRIDSP) 3-96
id operand (SEGINIT) 3-98
ID operand (STARTGEN) 3-101
ID operand (STATION) 3-106
ID operand (TRANPL) 3-114
id suboperand (STARTGEN CNM) 3-104
ILVS operand (ALACU) C-7
IND operand (DCATRTBL) 3-16
IND operand (INTRTBL) 3-76
INDXC operand (FILES) 3-69
INDXn operand (FILES) 3-69
inserting diskettes 4-24
installation 4-6
installation diskette 4-4
INSTR operand (STATION) 3-110
interfaces to LCF 4-41
INTMR operand (STATION) 3-111
INTRTBL macro 3-73
INTRTBL operand (TRTBHDR) 3-116
ISTATUS operand (ALALINE) C-3

J

JUCC suboperand (DEV4704 OPTIONS) 3-54

K

KEY operand (DEV3616) 3-49
KEY operand (DEV4710) 3-57
KEY operand (DEV4720) 3-60
keyboard, 50-key numeric 4-4
KEYED operand (FILES) 3-72
keywords
 BIND (STATION CPU) 3-110
 NOAUTO (STATION CPU) 3-110
 NOPURGE (STATION CPU) 3-110
KYBD operand (INTRTBL) 3-76

L

labeldca operand (DCAPORT) 3-14
labeldev operand (DEFADDR) 3-28
LAM operand (ALALINE) C-3
LCF diskette menu 4-6
LCF library diskette 4-2
LCF program diskette 4-2
LCF program menu 4-35
LCNTRL operand (STARTGEN) C-10
LDA macro 3-77
lda operand (LDA) 3-78
lda suboperand (STATION TERM) 3-107
letter operand (DKATBL) 3-62
library diskette 4-2, 4-3, 4-11, 4-14
limitations of LCF 4-46
LINE operand (ALACU) C-6
LINE operand (DEV3616) 3-50
LINE operand (DEV4710) 3-57
LINE operand (DEV4720) 3-60
LINELN operand (ALACU) C-6
lines suboperand (DEV3616 OFFSET) 3-49
LNKTB operand (ALALINE) C-4
LOG operand (ALACU) C-7
LOGTM operand (FILES) 3-72
loop operand (EXTCTR) 3-66
looplabel operand (DEFADDR) 3-29
LOOPS macro 3-80
LOOPS statement 4-17
LPI operand (DEV3616) 3-50
LPI operand (DEV4720) 3-60
LUASSIGN operand (COMLINK) 3-11
LUSENSE operand (COMLINK) 3-11

M

m operand (COMLINK) 3-12
macro instructions, configuration 1-1
macros
 STARTGEN 3-101
main LCF menu 4-36
MARGIN operand (DEV3616) 3-50
mask operand (INTRTBL) 3-76
MAXSEG operand (STATION) 3-112
MAXSTOR operand (STATION) 3-109
MB operand (FILES) 3-72
menu, LCF diskette 4-6
menu, main LCF 4-36
messages A-1, B-1

mod operand (DEFADDR) 3-29
 MODEL operand (DCA3278) 3-18
 MODEL operand (DCA3279) 3-20
 MODEL operand (DCA4704) 3-25
 MODEL operand (DEV3604) 3-32
 MODEL operand (DEV3610) 3-38
 MODEL operand (DEV3611) 3-40
 MODEL operand (DEV3612) 3-42
 MODEL operand (DEV3615) 3-46
 MODEL operand (DEV4704) 3-53
 MODEL operand (DEV4720) 3-59
 module 3-84
 MON operand (APMEDIA) 3-4
 MONITOR operand (STARTGEN) 3-102
 mounting diskettes 4-24
 mscode operand (MSTRTBL) 3-81
 MSGLITE operand (STARTGEN) 3-102
 MSLITE operand 3-55
 MSLITE operand (DEV3604) 3-33
 MSR operand (DCA3278) 3-19
 MSRE suboperand (DCA4704 FEATURE) 3-26
 MSTRTBL macro 3-81
 MSTRTBL operand (DCA3278) 3-18
 MSTRTBL operand (DCA3279) 3-21
 MSTRTBL operand (DCA4704) 3-25
 MSTRTBL operand (DEV3604) 3-33
 MSTRTBL operand (DEV3606) 3-35
 MSTRTBL operand (DEV3608) 3-36
 MSTRTBL operand (DEV4704) 3-54

N

n operand (COMLINK) 3-12
 n operand (DCATRTBL) 3-15
 N suboperand (DCA3279 CURSOR) 3-21
 N suboperand (DCA3279 ERTLS) 3-21
 N suboperand (DCA4704 CURSOR) 3-26
 N suboperand (DCA4704 ERTLS) 3-26
 N suboperand (DEV3604 CURSOR) 3-33
 N suboperand (DEV3604 ERTLS) 3-34
 N suboperand (DEV4704 CURSOR) 3-54
 N suboperand (DEV4704 ERTLS) 3-55
 N suboperand (DEV4720 ADVANCE) 3-61
 N suboperand (PRIDSP STARTUP) 3-95
 name 3-97
 name operand (OPTMOD) 3-85
 name, program 4-25
 NCSR suboperand (DEV4704 OPTIONS) 3-54
 NETID operand (ALACU) C-6
 NETID operand (ALALINE) C-3
 NETID operand (ALATERM) C-8
 NL operand (DEV3616) 3-49
 NL operand (DEV4710) 3-57
 NL operand (DEV4720) 3-60
 NOALLOC operand (FILES) 3-71
 NOAUTO keyword 3-110
 NOEXP operand (COMLINK) 3-11
 NOKEYED operand (FILES) 3-71
 NOMB operand (FILES) 3-72
 NOOP operand (PINTBL) 3-94
 NOOP suboperand (MSTRTBL func) 3-82
 NOPURGE keyword 3-110
 NOPURGE operand (COMLINK) 3-11
 NOXLDKT operand (FILES) 3-72
 NOXRCD operand (FILES) 3-71
 NSCC operand (DCA3278) 3-19
 NSCC suboperand (DCA3279 OPTIONS) 3-21
 NSCC suboperand (DCA4704 OPTIONS) 3-26
 NSCC suboperand (DEV3604 OPTIONS) 3-33

NSCC suboperand (DEV4704 OPTIONS) 3-54
 NSML operand (DCA3278) 3-19
 NSML suboperand (DCA3279 OPTIONS) 3-21
 NSML suboperand (DCA4704 OPTIONS) 3-26
 NSML suboperand (DEV3604 OPTIONS) 3-33
 numeric keyboard 4-4

O

OCR suboperand (DEV3608 OPTIONS) 3-37
 OFF suboperand (PRIDSP STARTUP) 3-95
 OFFSET operand (DEV3608) 3-37
 OFFSET operand (DEV3611) 3-40
 OFFSET operand (DEV3612) 3-43
 OFFSET operand (DEV3616) 3-49
 OFFSET operand (DEV4710) 3-56
 OFFSET operand (DEV4720) 3-60
 ON suboperand (PRIDSP STARTUP) 3-95
 operands 3-2
 A (COMLINK) 3-9
 A (LDA) 3-78
 A (STATION) 3-108
 AA (COMLINK) 3-10
 AC (COMLINK) 3-10
 ACB (ALALINE) C-4
 ACB (COMLINK) 3-8
 accent (DKATBL) 3-62
 ADDR (COMLINK) 3-12
 ADVANCE (DEV4720) 3-61
 ALA (STATION) C-11
 ALAMAX (ALACU) C-7
 ALAMAX (ALATERM) C-9
 ALAMAX (STATION) C-11
 ALITE (DEV4704) 3-54
 ALLOC (FILES) 3-71
 ALT (DCATRTBL) 3-15
 ALTLED (FILES) 3-72
 ALTTBL (TRTLIST) 3-118
 AMS (STATION) C-11
 APBNM (STATION) 3-106
 APL (DCA3278) 3-19
 apname (APLIST) 3-3
 APSTACK (STATION) 3-112
 ASYNLDA (STATION) 3-111
 AVGML (DEV3608) 3-36
 B (COMLINK) 3-9
 B (LDA) 3-78
 B (STATION) 3-108
 BIND (COMLINK) 3-11
 BPS (LOOPS) 3-80
 BSC (COMLINK) 3-12
 BSC (STARTGEN) 3-104
 BSC (STATION) 3-111
 BSCOPT (COMLINK) 3-12
 BUF (FILES) 3-70
 BUFFER (ALALINE) C-4
 buffers (FILES) 3-71
 CANCEL (TRTBHDR) 3-116
 CASE (DCATRTBL) 3-16
 CASE (INTRTBL) 3-76
 CFOLD (DEV3611) 3-40
 CFOLD (DEV3612) 3-43
 CFOLD (DEV3616) 3-49
 CHAIN (DCA3262) 3-17
 char (INTRTBL) 3-74
 char (MSTRTBL) 3-81
 CHARDEF (OUTBHDR) 3-87
 CHARSET (DEV4704) 3-53
 CHARSET (OUTRTBL) 3-88

CHARSET (OUTSPEC) 3-91
 C'c' (PINTBL) 3-93
 CLASS (SEGSTOR) 3-100
 CNB (ALABUFF) C-2
 CNB (COMLINK) 3-8
 CNL (ALABUFF) C-2
 CNL (COMLINK) 3-7
 CNM (STARTGEN) 3-104
 code (INTRTBL) 3-73
 COMF (FILES) 3-69
 comp (EXTCTR) 3-66
 component (LDA) 3-77
 composite (DKATBL) 3-62
 CONSHR (DEV3610) 3-38
 CONSHR (DEV3612) 3-43
 CONSHR (DEV3615) 3-46
 CONSHR (DEV3616) 3-49
 CONV (COMLINK) 3-13
 CPGID (OUTSPEC) 3-91
 CPI (DEV3616) 3-50
 CPI (DEV4710) 3-57
 CPI (DEV4720) 3-60
 CPU (STATION) 3-110
 CPYAP (FILES) 3-70
 CTG (ALALINE) C-4
 CTG (COMLINK) 3-9
 CTLR (STARTGEN) 3-101
 CU (ALATERM) C-8
 CURSOR (DCA3278) 3-18
 CURSOR (DCA3279) 3-21
 CURSOR (DCA4704) 3-26
 CURSOR (DEV3604) 3-33
 CURSOR (DEV4704) 3-54
 DA (STATION) 3-109
 DASD (COMLINK) 3-12
 data (ENDGEN) 3-65
 data (SEGINIT) 3-99
 datasets (FILES) 3-70
 DATE (STARTGEN) 3-101
 DBA (ALALINE) C-5
 DC (COMLINK) 3-10
 DCE (COMLINK) 3-8
 DCL (ALALINE) C-5
 DCL (COMLINK) 3-7
 DEBLSE (DEV3610) 3-39
 DEBLSE (DEV3612) 3-43
 DEF (COMLINK) 3-12
 DEF (OUTRTBL) 3-88
 defaddr (LDA) 3-77
 DEFCHAR (DEV3616) 3-50
 DEFCHAR (DEV4710) 3-57
 DEFCHAR (DEV4720) 3-61
 DEFQUAL (DEV4720) 3-61
 DEFTRT (DEV3616) 3-50
 DEFTRT (DEV4710) 3-57
 DEFTRT (DEV4720) 3-61
 DELSET (STATION) 3-109
 DEL1 (COMLINK) 3-10
 DEL2 (COMLINK) 3-10
 DEVICE (CHARDEF) 3-6
 DEVICE (DCATRTBL) 3-15
 device (LDA) 3-77
 DEVICE (OUTRTBL) 3-88
 DEVICE (OUTSPEC) 3-91
 disp (ENDGEN) 3-64
 disp (SEGINIT) 3-98
 DKATBL (TRTBHDR) 3-116
 DKT (FILES) 3-68
 DPOOL (STATION) 3-113
 DRBC (COMLINK) 3-12
 DSK (FILES) 3-68
 DSKOP (STARTGEN) 3-102
 DSL (COMLINK) 3-8
 DSM (STATION) 3-111
 DTACC (STATION) 3-113
 DUMP (STARTGEN) 3-103
 EDAM (FILES) 3-70
 EID (INTRTBL) 3-76
 ENCRYPT (PINTBL) 3-93
 END (PINTBL) 3-93
 EOF (INTRTBL) 3-75
 EOFA (INTRTBL) 3-75
 EOFB (INTRTBL) 3-75
 EOM (INTRTBL) 3-75
 EOM (PINTBL) 3-94
 EOMA (INTRTBL) 3-75
 EOMB (INTRTBL) 3-75
 EOMSET (DCA3278) 3-18
 EOMSET (DCA3279) 3-20
 EOMSET (DCA4704) 3-26
 EOMSET (DEV3604) 3-33
 EOMSET (DEV4704) 3-54
 ERTLS (DCA3278) 3-19
 ERTLS (DCA3279) 3-21
 ERTLS (DCA4704) 3-26
 ERTLS (DEV3604) 3-34
 ERTLS (DEV4704) 3-55
 FEATURE (DCA3278) 3-19
 FEATURE (DCA3279) 3-22
 FEATURE (DCA4704) 3-26
 FILL (PINTBL) 3-93
 FLAG (COMLINK) 3-10
 FONT (DEV4720) 3-60
 FORMWD (DEV3608) 3-36
 FT (DEV3610) 3-38
 FT (DEV3612) 3-43
 FT (DEV3615) 3-46
 func (INTRTBL) 3-74
 func (MSTRTBL) 3-82
 func (OPTMOD) 3-85
 gpc (COMLINK) 3-12
 ID (DEFSPA) 3-31
 id (ENDGEN) 3-64
 ID (EXTCTR) 3-67
 ID (LOOPS) 3-80
 ID (PRIDSP) 3-96
 id (SEGINIT) 3-98
 ID (STARTGEN) 3-101
 ID (STATION) 3-106
 ID (TRANPL) 3-114
 ILVS (ALACU) C-7
 IND (DCATRTBL) 3-16
 IND (INTRTBL) 3-76
 INDXC (FILES) 3-69
 INDXn (FILES) 3-69
 INSTR (STATION) 3-110
 INTMR (STATION) 3-111
 INTRTBL (TRTBHDR) 3-116
 ISTATUS (ALALINE) C-3
 KEY (DEV3616) 3-49
 KEY (DEV4710) 3-57
 KEY (DEV4720) 3-60
 KEYED (FILES) 3-72
 KYBD (INTRTBL) 3-76
 labeldca (DCAPORT) 3-14
 labeldev (DEFADDR) 3-28
 LAM (ALALINE) C-3
 LCNTRL (STARTGEN) C-10
 lda (LDA) 3-78
 lda (STATION TERM) 3-107

letter (DKATBL) 3-62
LINE (ALACU) C-6
LINE (DEV3616) 3-50
LINE (DEV4710) 3-57
LINE (DEV4720) 3-60
LINELN (ALACU) C-6
LNKTB (ALALINE) C-4
LOG (ALACU) C-7
LOGTM (FILES) 3-72
loop (EXTCTR) 3-66
looplabel (DEFADDR) 3-29
LPI (DEV3616) 3-50
LPI (DEV4720) 3-60
LUASSIGN (COMLINK) 3-11
LUSENSE (COMLINK) 3-11
m (COMLINK) 3-12
MARGIN (DEV3616) 3-50
mask (INTRTBL) 3-76
MAXSEG (STATION) 3-112
MAXSTOR (STATION) 3-109
MB (FILES) 3-72
mod (DEFADDR) 3-29
MODEL (DCA3278) 3-18
MODEL (DCA3279) 3-20
MODEL (DCA4704) 3-25
MODEL (DEV3604) 3-32
MODEL (DEV3610) 3-38
MODEL (DEV3611) 3-40
MODEL (DEV3612) 3-42
MODEL (DEV3615) 3-46
MODEL (DEV4704) 3-53
MODEL (DEV4720) 3-59
module 3-84
MON (APMEDIA) 3-4
MONITOR (STARTGEN) 3-102
mscode (MSTRTBL) 3-81
MSGLITE (STARTGEN) 3-102
MSLITE (DEV3604) 3-33
MSR (DCA3278) 3-19
MSTRTBL (DCA3278) 3-18
MSTRTBL (DCA3279) 3-21
MSTRTBL (DCA4704) 3-25
MSTRTBL (DEV3604) 3-33
MSTRTBL (DEV3606) 3-35
MSTRTBL (DEV3608) 3-36
MSTRTBL (DEV4704) 3-54
n (COMLINK) 3-12
n (DCATRTBL) 3-15
name 3-97
name (OPTMOD) 3-85
NETID (ALACU) C-6
NETID (ALALINE) C-3
NETID (ALATERM) C-8
NL (DEV3616) 3-49
NL (DEV4710) 3-57
NL (DEV4720) 3-60
NOALLOC (FILES) 3-71
NOEXP (COMLINK) 3-11
NOKEYED (FILES) 3-71
NOMB (FILES) 3-72
NOOP (PINTBL) 3-94
NOPURGE (COMLINK) 3-11
NOXLDKT (FILES) 3-72
NOXRCD (FILES) 3-71
NSCC (DCA3278) 3-19
NSML (DCA3278) 3-19
OFFSET (DEV3608) 3-37
OFFSET (DEV3611) 3-40
OFFSET (DEV3612) 3-43
OFFSET (DEV3616) 3-49
OFFSET (DEV4710) 3-56
OFFSET (DEV4720) 3-60
OPTIONS (COMLINK) 3-11
OPTIONS (DCA3278) 3-19
OPTIONS (DCA3279) 3-21
OPTIONS (DCA4704) 3-26
OPTIONS (DEV3604) 3-33
OPTIONS (DEV3608) 3-37
OPTIONS (DEV3610) 3-39
OPTIONS (DEV3611) 3-41
OPTIONS (DEV3612) 3-43
OPTIONS (DEV3615) 3-47
OPTIONS (DEV4704) 3-54
OUTBHDR (DEV3616) 3-48
OUTBHDR (DEV4720) 3-59
OUTRTBL (DCA3278) 3-18
OUTRTBL (DCA3279) 3-20
OUTRTBL (DCA4704) 3-25
OUTRTBL (DEV3604) 3-32
OUTRTBL (DEV3606) 3-35
OUTRTBL (DEV3608) 3-36
OUTRTBL (DEV3610) 3-38
OUTRTBL (DEV3611) 3-40
OUTRTBL (DEV3612) 3-42
OUTRTBL (DEV3615) 3-46
OUTRTBL (DEV4704) 3-53
OUTSPEC (OUTBHDR) 3-87
PACE (COMLINK) 3-11
PACING (ALATERM) C-9
PBFOLD (DEV3616) 3-50
PF (FILES) 3-69
pincode (PINTBL) 3-93
PINTBL (DCA4704) 3-26
PINTBL (DEV3604) 3-34
PINTBL (DEV4704) 3-54
POLL (ALACU) C-6
poolname (DCAPORT) 3-14
poolname (DEFADDR) 3-28
PORT (ALALINE) C-5
PORT (DCAPORT) 3-14
port (STATION DA) 3-109
pos (OUTRTBL) 3-88
pos (OUTSPEC) 3-90
PRINT (STARTGEN) 3-103
PRT (COMLINK) 3-8
PRTY (DEV3604) 3-34
PRTY (DEV3606) 3-35
PRTY (DEV3608) 3-37
PRTY (DEV3610) 3-39
PRTY (DEV3611) 3-41
PRTY (DEV3612) 3-44
PRTY (DEV3615) 3-47
PRTY (DEV3616) 3-51
PRTY (DEV4704) 3-53
PRTY (DEV4710) 3-57
PRTY (DEV4720) 3-61
PS (DCA3262) 3-17
PS (DCA3287) 3-23
PS (DCA5210) 3-24
PS (DEV3610) 3-38
PS (DEV3611) 3-40
PS (DEV3612) 3-42
PS (DEV3615) 3-46
PS (DEV3616) 3-48
PS (DEV4710) 3-56
PS (DEV4720) 3-59
PSFC (ALALINE) C-4
PUNID (ALACU) C-7
PUTYPE (ALACU) C-7
R (APLIST) 3-3

R (DCA3279) 3-21
RETRIES (COMLINK) 3-10
RETRY (ALACU) C-7
RETRY (ALALINE) C-4
RETSTK (STATION) 3-111
RFSH (TRANPL) 3-115
RSP (ALATERM) C-9
r1,c1 (DEFSBF) 3-30
sa (DEFADDR) 3-28
sa (EXTCTR) 3-67
scancode (DCATRTBL) 3-15
SCRANGE (INTRTBL) 3-76
seg (ENDGEN) 3-64
seg (SEGINIT) 3-98
SEGALL (STATION) 3-113
SEGMENT (ALATERM) C-9
SEGSIZE (SEGSTOR) 3-100
SEL (ALATERM) C-8
SELSEQ (COMLINK) 3-10
SHARED (DEV3610) 3-38
SHARED (DEV3611) 3-41
SHARED (DEV3612) 3-43
SHARED (DEV3615) 3-46
SHARED (DEV3616) 3-49
SHARED (DEV4710) 3-56
SHARED (DEV4720) 3-60
SHARED (STATION) 3-110
SHIFT (TRTBHDR) 3-117
SHIFT (DCATRTBL) 3-15
SHIFT1 (TRTLIST) 3-118
SHIFT2 (TRTLIST) 3-118
SHIFT3 (TRTLIST) 3-118
SHIFT4 (TRTLIST) 3-118
SIZE (DEFSPA) 3-31
size (TRANPL) 3-114
size,count (TRANPL) 3-114
SKEW (DEV4720) 3-61
SLOW (ALALINE) C-4
SS (STATION) 3-109
STA (ALATERM) C-8
START (PINTBL) 3-93
STARTUP (PRIDSP) 3-95
STARTUP (STATION) 3-110
STATS (STARTGEN) 3-103
stid (PRIDSP) 3-95
SWITCH (DCA3262) 3-17
SWITCH (DCA3287) 3-23
SWITCH (DCA5210) 3-24
SWM (COMLINK) 3-8
S13 (SEGSTOR) 3-100
S13 (STARTGEN) 3-101
S131D (SEGSTOR) 3-100
S15 (STARTGEN) 3-102
T (APLIST) 3-3
term (STATION TERM) 3-107
TERM (STATION) 3-107
TF (FILES) 3-69
TIMEOUT (DEV3610) 3-38
TIMEOUT (DEV3612) 3-43
TIMEOUT (STARTGEN) 3-102
tla (DEFADDR) 3-29
tla (EXTCTR) 3-66
TM (DCA3262) 3-17
TM (DCA3287) 3-23
TM (DCA5210) 3-24
TOALERT (STARTGEN) 3-102
TRACE (STARTGEN) 3-105
TRANPL (STATION) 3-113
TRTBHDR (DCA3278) 3-18
TRTBHDR (DCA3279) 3-20
TRTBHDR (DCA4704) 3-25
TRTBHDR (DEV3604) 3-32
TRTBHDR (DEV3606) 3-35
TRTBHDR (DEV3608) 3-36
TRTBHDR (DEV4704) 3-53
TRTLIST (DCA4704) 3-25
TYPA (TRTBHDR) 3-116
TYPE (COMLINK) 3-9
TYPE (LOOPS) 3-80
TYPE (MSTRTBL) 3-82
UAP (APMEDIA) 3-4
UDDccc (DCA3278) 3-19
UDDccc (DCA3279) 3-21
UDDccc (DCA4704) 3-27
UDDccc (DEV3604) 3-34
UDDccc (DEV3608) 3-37
UDDccc (DEV3612) 3-44
UDDccc (DEV3616) 3-51
UDDccc (DEV4704) 3-55
UDDCTF (DEV3614) 3-45
UDDCTF (DEV3624) 3-52
UDDFST (DEV3606) 3-35
UDDPC1 (DCA3262) 3-17
UDDPC1 (DCA3287) 3-23
UDDPC1 (DCA5210) 3-24
UDDPC1 (DEV3610) 3-39
UDDPC1 (DEV3611) 3-41
UDDPC1 (DEV3615) 3-47
UDDPC1 (DEV4710) 3-58
UDDPC1 (DEV4720) 3-61
VERSION (COMLINK) 3-9
vn (APLIST) 3-3
VN (STARTGEN) 3-101
WACK (COMLINK) 3-13
WL (DEV3610) 3-38
WL (DEV3612) 3-42
WL (DEV3615) 3-46
WL (DEV3616) 3-48
WL (DEV4710) 3-56
WL (DEV4720) 3-59
WRAP (ALALINE) C-5
WRT (COMLINK) 3-8
WRT (STATION) 3-110
X'aa' (OUTSPEC) 3-91
X'nn' (PINTBL) 3-93
X'n' (OUTRTBL) 3-88
XLDKT (FILES) 3-72
XRCD (FILES) 3-71
1422 (COMLINK) 3-9
4502 (COMLINK) 3-9
4850 (COMLINK) 3-9
5656 (COMLINK) 3-9
9494 (DEV3614) 3-45
9495 (DEV3614) 3-45
operating diskette 4-4, 4-22
OPTIONS operand (COMLINK) 3-11
OPTIONS operand (DCA3278) 3-19
OPTIONS operand (DCA3279) 3-21
OPTIONS operand (DCA4704) 3-26
OPTIONS operand (DEV3604) 3-33
OPTIONS operand (DEV3608) 3-37
OPTIONS operand (DEV3610) 3-39
OPTIONS operand (DEV3611) 3-41
OPTIONS operand (DEV3612) 3-43
OPTIONS operand (DEV3615) 3-47
OPTIONS operand (DEV4704) 3-54
OPTLIST macro 3-84
OPTMOD macro 3-85
OUTBHDR macro 3-87
OUTBHDR operand (DEV3616) 3-48

OUTBHDR operand (DEV4720) 3-59
 OUTRTBL macro 3-88
 OUTRTBL operand (DCA3278) 3-18
 OUTRTBL operand (DCA3279) 3-20
 OUTRTBL operand (DCA4704) 3-25
 OUTRTBL operand (DEV3604) 3-32
 OUTRTBL operand (DEV3606) 3-35
 OUTRTBL operand (DEV3608) 3-36
 OUTRTBL operand (DEV3610) 3-38
 OUTRTBL operand (DEV3611) 3-40
 OUTRTBL operand (DEV3612) 3-42
 OUTRTBL operand (DEV3615) 3-46
 OUTRTBL operand (DEV4704) 3-53
 OUTSPEC macro 1-10, 3-90
 OUTSPEC operand (OUTBHDR) 3-87

P

P command 4-14
 PACE operand (COMLINK) 3-11
 PACING operand (ALATERM) C-9
 panels 4-35
 PBFOLD operand (DEV3616) 3-50
 PF operand (FILES) 3-69
 pincode operand (PINTBL) 3-93
 PINTBL macro 3-93
 PINTBL operand (DCA4704) 3-26
 PINTBL operand (DEV3604) 3-34
 PINTBL operand (DEV4704) 3-54
 POLL operand (ALACU) C-6
 poolname operand (DCAPORT) 3-14
 poolname operand (DEFADDR) 3-28
 PORT operand (DCAPORT) 3-14
 PORT operand (ALALINE) C-5
 port suboperand (STATION DA) 3-109
 pos operand (OUTRTBL) 3-88
 pos operand (OUTSPEC) 3-90
 pre-operating diskette 4-4
 prerequisites 4-4
 PRIDISP macro 3-95
 PRINT operand (STARTGEN) 3-103
 program menu, LCF 4-35
 program transfer panel 4-39
 program, host application 4-41
 programmed symbol sets 3-97
 programs via diskette 4-25
 programs, allocating space for 4-26
 programs, maximum number 4-27
 programs, replacing and adding 4-25
 protocols, SNA 4-43
 PRT operand (COMLINK) 3-8
 PRTY operand (DEV3604) 3-34
 PRTY operand (DEV3606) 3-35
 PRTY operand (DEV3608) 3-37
 PRTY operand (DEV3610) 3-39
 PRTY operand (DEV3611) 3-41
 PRTY operand (DEV3612) 3-44
 PRTY operand (DEV3615) 3-47
 PRTY operand (DEV3616) 3-51
 PRTY operand (DEV4704) 3-53
 PRTY operand (DEV4710) 3-57
 PRTY operand (DEV4720) 3-61
 PS operand (DCA3262) 3-17
 PS operand (DCA3287) 3-23
 PS operand (DCA5210) 3-24
 PS operand (DEV3610) 3-38
 PS operand (DEV3611) 3-40
 PS operand (DEV3612) 3-42
 PS operand (DEV3615) 3-46

PS operand (DEV3616) 3-48
 PS operand (DEV4710) 3-56
 PS operand (DEV4720) 3-59
 PSFC operand (ALALINE) C-4
 PSSLST macro 1-10, 3-97
 PUNID operand (ALACU) C-7
 PUTYPE operand (ALACU) C-7

Q

Q suboperand (DEV4720 FONT) 3-60

R

R operand (APLIST) 3-3
 R operand (DCA3279) 3-21
 R suboperand (DCA4704 CURSOR) 3-26
 R suboperand (DEV3604 CURSOR) 3-33
 R suboperand (DEV4704 CURSOR) 3-54
 R suboperand (DEV4720 FONT) 3-60
 receive CPGEN panel 4-40
 receiving CPGEN files 4-33
 replace configuration panel 4-37
 replacing programs 4-25
 restrictions 4-46
 RETRIES operand (COMLINK) 3-10
 RETRY operand (ALACU) C-7
 RETRY operand (ALALINE) C-4
 RETSTK operand (STATION) 3-111
 RFSH operand (TRANPL) 3-115
 RSP operand (ALATERM) C-9
 r1,c1 operand (DEFSBF) 3-30

S

S command 4-14, 4-22
 sa operand (DEFADDR) 3-28
 sa operand (EXTCTR) 3-67
 SBF suboperand (STATION DSM) 3-112
 scancode operand (DCATRTBL) 3-15
 SCRANGE operand (INTRTBL) 3-76
 seg operand (ENDGEN) 3-64
 seg operand (SEGINIT) 3-98
 seg suboperand (STATION AMS) C-11
 SEGALL operand (STATION) 3-113
 SEGINIT macro 3-98
 SEGMENT operand (ALATERM) C-9
 SEGSIZE operand (SEGSTOR) 3-100
 SEGSTOR macro 3-100
 SEL operand (ALATERM) C-8
 SELSEQ operand (COMLINK) 3-10
 SHARED operand (DEV3610) 3-38
 SHARED operand (DEV3611) 3-41
 SHARED operand (DEV3612) 3-43
 SHARED operand (DEV3615) 3-46
 SHARED operand (DEV3616) 3-49
 SHARED operand (DEV4710) 3-56
 SHARED operand (DEV4720) 3-60
 SHARED operand (STATION) 3-110
 SHFT operand (TRTBHDR) 3-117
 SHIFT operand (DCATRTBL) 3-15
 SHIFT1 operand (TRTLIST) 3-118
 SHIFT2 operand (TRTLIST) 3-118
 SHIFT3 operand (TRTLIST) 3-118
 SHIFT4 operand (TRTLIST) 3-118
 SIZE operand (DEFSPA) 3-31

size operand (TRANPL) 3-114
 size, CPGEN 4-45
 size,count operand (TRANPL) 3-114
 SKEW operand (DEV4720) 3-61
 skip suboperand (DEV3616 CFOLD) 3-49
 SL command 4-14
 SLOW operand (ALALINE) C-4
 SNA protocols 4-43
 SPECIFY suboperand (DEV3616 NL) 3-49
 SPECIFY suboperand (DEV4720 NL) 3-60
 SR command 4-14
 SS operand (STATION) 3-109
 ssid suboperand (STATION AMS) C-11
 ST command 4-14
 STA operand (ALATERM) C-8
 START operand (PINTBL) 3-93
 STARTGEN C-10
 STARTGEN macro 3-101
 starting LCF 4-6
 startup 4-2
 STARTUP operand (PRIDSP) 3-95
 STARTUP operand (STATION) 3-110
 statement entry panel 4-37
 statements, correcting 4-15
 statements, CPGEN 4-1
 STATION C-11
 STATION macro 3-106
 STATS operand (STARTGEN) 3-103
 steps suboperand (DEV3616 OFFSET) 3-49
 stid operand (PRIDSP) 3-95
 suboperands
 APL (DCA3279 FEATURE) 3-22
 AUTO (DEV3616 NL) 3-49
 AUTO (DEV4720 NL) 3-60
 begin (DEV3616 CFOLD) 3-49
 DCA (STATION DSM) 3-112
 DEF (STATION BSC) 3-111
 EAB (STATION DSM) 3-112
 EID (MSTRTBL func) 3-82
 EOF (MSTRTBL func) 3-82
 EOM (MSTRTBL func) 3-82
 id (STARTGEN CNM) 3-104
 JUCC (DEV4704 OPTIONS) 3-54
 lines (DEV3616 OFFSET) 3-49
 MSRE (DCA4704 FEATURE) 3-26
 N (DCA3279 CURSOR) 3-21
 N (DCA3279 ERTLS) 3-21
 N (DCA4704 CURSOR) 3-26
 N (DCA4704 ERTLS) 3-26
 N (DEV3604 CURSOR) 3-33
 N (DEV3604 ERTLS) 3-34
 N (DEV4704 CURSOR) 3-54
 N (DEV4704 ERTLS) 3-55
 N (DEV4720 ADVANCE) 3-61
 N (PRIDSP STARTUP) 3-95
 NCSR (DEV4704 OPTIONS) 3-54
 NOOP (MSTRTBL func) 3-82
 NSCC (DCA3279 OPTIONS) 3-21
 NSCC (DCA4704 OPTIONS) 3-26
 NSCC (DEV3604 OPTIONS) 3-33
 NSCC (DEV4704 OPTIONS) 3-54
 NSML (DCA3279 OPTIONS) 3-21
 NSML (DCA4704 OPTIONS) 3-26
 NSML (DEV3604 OPTIONS) 3-33
 OCR (DEV3608 OPTIONS) 3-37
 OFF (PRIDSP STARTUP) 3-95
 ON (PRIDSP STARTUP) 3-95
 Q (DEV4720 FONT) 3-60
 R (DCA4704 CURSOR) 3-26
 R (DEV3604 CURSOR) 3-33

R (DEV4704 CURSOR) 3-54
 R (DEV4720 FONT) 3-60
 SBF (STATION DSM) 3-112
 seg (STATION AMS) C-11
 skip (DEV3616 CFOLD) 3-49
 SPECIFY (DEV3616 NL) 3-49
 SPECIFY (DEV4720 NL) 3-60
 ssid (STATION AMS) C-11
 steps (DEV3616 OFFSET) 3-49
 term (STARTGEN CNM) 3-104
 tranpl (STARTGEN CNM) 3-104
 usc (STARTGEN BSC) 3-104
 usc (STATION BSC) 3-111
 Y (DCA3279 CURSOR) 3-21
 Y (DCA3279 ERTLS) 3-21
 Y (DCA4704 CURSOR) 3-26
 Y (DCA4704 ERTLS) 3-26
 Y (DEV3604 CURSOR) 3-33
 Y (DEV3604 ERTLS) 3-34
 Y (DEV4704 CURSOR) 3-54
 Y (DEV4704 ERTLS) 3-55
 Y (DEV4720 ADVANCE) 3-61
 1586 (DEV3604 OPTIONS) 3-33
 4905 (DEV3604 OPTIONS) 3-33
 4905 (DEV4704 OPTIONS) 3-54
 9641 (DEV3604 OPTIONS) 3-33
 9641 (DEV4704 OPTIONS) 3-54
 SWITCH operand (DCA3262) 3-17
 SWITCH operand (DCA3287) 3-23
 SWITCH operand (DCA5210) 3-24
 SWM operand (COMLINK) 3-8
 symbol sets, programmed 3-97
 S13 operand (SEGSTOR) 3-100
 S13 operand (STARTGEN) 3-101
 S131D operand (SEGSTOR) 3-100
 S15 operand (STARTGEN) 3-102

T

T operand (APLIST) 3-3
 TERM operand (STATION) 3-107
 term suboperand (STARTGEN CNM) 3-104
 term suboperand (STATION TERM) 3-107
 terminating the file 4-11
 TF operand (FILES) 3-69
 TIMEOUT operand (DEV3610) 3-38
 TIMEOUT operand (DEV3612) 3-43
 TIMEOUT operand (STARTGEN) 3-102
 tla operand (DEFADDR) 3-29
 tla operand (EXTCTR) 3-66
 TM operand (DCA3262) 3-17
 TM operand (DCA3287) 3-23
 TM operand (DCA5210) 3-24
 TOALERT operand (STARTGEN) 3-102
 TRACE operand (STARTGEN) 3-105
 tracks 4-27
 TRANPL macro 3-114
 TRANPL operand (STATION) 3-113
 tranpl suboperand (STARTGEN CNM) 3-104
 transfer application panel 4-39
 transferring CPGEN files 4-3
 transferring programs 4-3
 transmission sequences 4-41
 TRTBHDR macro 3-116
 TRTBHDR operand (DCA3278) 3-18
 TRTBHDR operand (DCA3279) 3-20
 TRTBHDR operand (DCA4704) 3-25
 TRTBHDR operand (DEV3604) 3-32
 TRTBHDR operand (DEV3606) 3-35

TRTBHDR operand (DEV3608) 3-36
TRTBHDR operand (DEV4704) 3-53
TRTLIST macro 1-11, 3-118
TRTLIST operand (DCA4704) 3-25
TYPA operand (TRTBHDR) 3-116
TYPE operand (COMLINK) 3-9
TYPE operand (LOOPS) 3-80
TYPE operand (MSTRTBL) 3-82

U

UAP operand (APMEDIA) 3-4
UDDccc operand (DCA3278) 3-19
UDDccc operand (DCA3279) 3-21
UDDccc operand (DCA4704) 3-27
UDDccc operand (DEV3604) 3-34
UDDccc operand (DEV3608) 3-37
UDDccc operand (DEV3612) 3-44
UDDccc operand (DEV3616) 3-51
UDDccc operand (DEV4704) 3-55
UDDCTF operand (DEV3614) 3-45
UDDCTF operand (DEV3624) 3-52
UDDFST operand (DEV3606) 3-35
UDDPC1 operand (DCA3262) 3-17
UDDPC1 operand (DCA3287) 3-23
UDDPC1 operand (DCA5210) 3-24
UDDPC1 operand (DEV3610) 3-39
UDDPC1 operand (DEV3611) 3-41
UDDPC1 operand (DEV4710) 3-58
UDDPC1 operand (DEV4720) 3-61
UPPDC1 operand (DEV3615) 3-47
usc suboperand (STARTGEN BSC) 3-104
usc suboperand (STATION BSC) 3-111
using LCF 4-1

V

VERSION operand (COMLINK) 3-9
vn operand (APLIST) 3-3
VN operand (STARTGEN) 3-101
VTAM 4-42

W

WACK operand (COMLINK) 3-13
WL operand (DEV3610) 3-38
WL operand (DEV3612) 3-42
WL operand (DEV3615) 3-46
WL operand (DEV3616) 3-48
WL operand (DEV4710) 3-56
WL operand (DEV4720) 3-59
WRAP operand (ALALINE) C-5

WRT operand (COMLINK) 3-8
WRT operand (STATION) 3-110

X

X'aa' operand (OUTSPEC) 3-91
X'nn' operand (PINTBL) 3-93
X'n' operand (OUTRTBL) 3-88
XLDKT operand (FILES) 3-72
XRCD operand (FILES) 3-71

Y

Y suboperand (DCA3279 CURSOR) 3-21
Y suboperand (DCA3279 ERTLS) 3-21
Y suboperand (DCA4704 CURSOR) 3-26
Y suboperand (DCA4704 ERTLS) 3-26
Y suboperand (DEV3604 CURSOR) 3-33
Y suboperand (DEV3604 ERTLS) 3-34
Y suboperand (DEV4704 CURSOR) 3-54
Y suboperand (DEV4704 ERTLS) 3-55
Y suboperand (DEV4720 ADVANCE) 3-61

1

1422 operand (COMLINK) 3-9
1586 suboperand (DEV3604 OPTIONS) 3-33

4

4502 operand (COMLINK) 3-9
4850 operand (COMLINK) 3-9
4905 suboperand (DEV3604 OPTIONS) 3-33
4905 suboperand (DEV4704 OPTIONS) 3-54

5

50-key keyboard 4-4
5656 operand (COMLINK) 3-9

9

9494 operand (DEV3614) 3-45
9495 operand (DEV3614) 3-45
9641 suboperand (DEV3604 OPTIONS) 3-33
9641 suboperand (DEV4704 OPTIONS) 3-54

4700 Finance Communication System
Controller Programming Library
Volume 6
Control Program Generation
Order No. GC31-2071-1

**READER'S
COMMENT
FORM**

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A.
(Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Reader's Comment Form

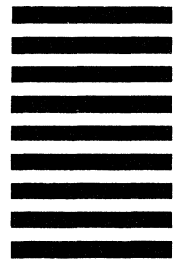
Fold and Tape

Please Do Not Staple

Fold and Tape



NO
POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED
STATES



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

**International Business Machines Corporation
Department 78C
1001 W.T. Harris Boulevard
Charlotte, NC, USA 28257**

Fold and Tape

Please Do Not Staple

Fold and Tape



File Number
S370/4300/8100/S34-30

Publication Number
GC31-2071-1

Printed in
USA

IBM