60459320

CONTROL DATA CORPORATION

NOS VERSION 2 INSTALLATION HANDBOOK

CDC® COMPUTER SYSTEMS:

CYBER 180

CYBER 170

CYBER 70

MODELS 71, 72, 73, 74

6000

REVISION RECORD

REVISION	DESCRIPTION
A (04-26-82)	Handbook released; reflects NOS 2.0 (which succeeds NOS 1.4) at PSR level 562/552 and supersedes the NOS Version 1 Installation Handbook. It documents the CYBER 170 Computer System, Models 825, 835, and 855, and 819 disks; includes a section on APRDECK entries; adds IOB, PPB, DOWN, ISHARE, and PRESET CMRDECK entries; drops documentation of ACCOUNT, AUTOROLL, BATCHIO, EI200, MAGNET, MSS MASTER, PRIORITY, QPROTECT, TELEX, USERS, and VALID IPRDECK entries; adds BIO, EXTENDED STACK PURGING, LOGGING, MAP, MASTER MSS, SECONDARY USER CARDS, and UNLOCK IPRDECK entries; and documents the changed format of IPRDECK entries. The handbook also documents the two new common disk area utility options (environment interface and microcode) and three new decks for NOS installation parameters (COMSMSC, COMSSFS, and COMSSRU). It drops documentation of decks COMPCMX, COMSEXP, and RESEQ, CPUMTR segmentation, and procedure file LIBMOD. The handbook drops documentation of Advanced Access Methods Version 1, ALGOL Version 4, COBOL Version 4, COBOL Version 4 to COBOL Version 5 Conversion Aids, CYBER Database Control System Version 1, Data Description Language Version 2, Time-Sharing Module, Export/Import Module, and TAF/TS Transaction Facility Data Manager. The handbook also drops documentation of the following equipment: 841, 65x, 6671/6676/2550-100.
B (01-27-83)	Revised to reflect NOS 2.1 at PSR summary level 580/577. This revision documents the CYBER 170 Computer System models 815, 865, and 875, and the 885-42 Disk Storage Subsystem; adds the Remote Diagnostic Facility and Remote Host Facility; revises installation procedures for NOS 2, Cross, CCP, and NAM; drops the use of procedures GENJOB and SYSTEM; adds procedures SETUP and MISCGET; adds production environment installation through use of a global library; adds LID, PRIVILEGED RDF, and RESIDENT RDF IPRDECK entries; adds short printer paper parameter to unit record EST entry; adds two-port multiplexer, network access device, and MAP III equipment EST entries; adds EST entry for the 885-42 disk; drops IOB and UEC CMRDECK entries; adds LIDT and XM CMRDECK entries; revises release tape contents; and includes editorial and technical corrections. Because of extensive changes, change bars and dots are not used. This edition obsoletes all previous editions.
C (10-11-83)	Revised to reflect NOS 2.2 at PSR summary level 596/587. This revision documents the installation procedures for the Full Screen Editor, Interactive Transfer Facility, and NOS-SCOPE 2 Station Facility. It documents changes to the CMRDECK and addition of the EQPDECK. This revision adds the following items for multilevel security: deck COMSMLS, CMRDECK entry OPSECM, EQPDECK entry ACCESS, and IPRDECK entries SECCATS, OQSH, and SECURES. It adds parameters for secure login; adds information for the multihost capability; describes IPRDECK entries COMLIB, EI, MICRO, MEMORY CLEARING, SPC, SPD, SPL, SPW, CLASS, and PCLASS; describes the modified MOVEPF utility; adds verification for RHF; documents CTI changes; and adds more description of DECKOPL parameters and procedures. This edition obsoletes all previous editions.
D (10-12-84)	Revised to reflect NOS 2.3 at PSR level 617/617. This revision documents the Tailored Release Process (TRP). Deadstart deck and CIP documentation have been deleted and moved to the Analysis Handbook and the CIP User's Handbook. Because of extensive changes, change bars and dots are not used and all pages reflect the current revision level. This edition obsoletes all previous editions.
E (03-22-85)	Revised to reflect NOS 2.4.1 at PSR level 630. This revision documents the CYBER 180 Computer Systems Models 840, 850, and 860, the 895 Disk Storage Subsystem, and installation information for Concurrent Maintenance Library (CML) and Mass Storage Archival Subsystem (MAS). This revision obsoletes all previous editions.
Publication No. 60459320	

REVISION LETTERS I, O, Q, S, X AND Z ARE NOT USED.

Address comments concerning this manual to:

Control Data Corporation
Publications and Graphics Division
4201 North Lexington Avenue
St. Paul, Minnesota 55112

or use Comment Sheet in the back of this manual.

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual, are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

PAGE	REV
Front Cover	-
Title Page	-
2	E
3	E
4	E
5	E
6	E
7	E
8	E
9	E
10	E
11/12	E
13	E
14	E
15	E
16	E
17	E
Divider	-
1-1	D
1-2	E
1-3	E
Divider	-
2-1	E
2-2	D
2-3	E
2-4	E
2-5	E
2-6	E
2-7	E
2-8	E
2-9	E
Divider	-
3-1	E
3-2	D
3-3	E
3-4	E
3-5	D
3-6	E
3-7	E
Divider	-
4-1	D
4-2	E
4-3	E
4-4	E
4-5	E
4-6	E
4-7	E
4-8	E
4-9	E
4-10	E
4-11	E
4-12	E
4-13	E
4-14	E
4-15	E
4-16	E
4-17	E
4-18	E
4-19	E
4-20	E
4-21	E
4-22	E
4-23	E
Divider	-
5-1	E
5-2	E
5-3	E
5-4	E
5-5	E
5-6	E
5-7	E
5-8	E
5-9	E
5-10	E
5-11	E
5-12	E
5-13	E
5-14	E
5-15	E
5-16	E
5-17	E
5-18	E
5-19	E
5-20	E
5-21	E
5-22	E
5-23	E
5-24	E
5-25	E
5-26	E
5-27	E
5-28	E
5-29	E
5-30	E
5-31	E
5-32	E
5-33	E
5-34	E
5-35	E
5-36	E
5-37	E
5-38	E
5-39	E
5-40	E
5-41	E
5-42	E
5-43	E
5-44	E
5-45	E
5-46	E
5-47	E
5-48	E
5-49	E
5-50	E
5-51	E
5-52	E
5-53	E
5-54	E
5-55	E
5-56	E
5-57	E
5-58	E
5-59	E
5-60	E
5-61	E
5-62	E
5-63	E
5-64	E
5-65	E
5-66	E
5-67	E
5-68	E
5-69	E
5-70	E
5-71	E
5-72	E
5-73	E
5-74	E
5-75	E
5-76	E
5-77	E
5-78	E
5-79	E
5-80	E
5-81	E
5-82	E
5-83	E
5-84	E
5-85	E
5-86	E
5-87	E
5-88	E
5-89	E
5-90	E
5-91	E
5-92	E
5-93	E
5-94	E
5-95	E
5-96	E
5-97	E
5-98	E
5-99	E
5-100	E
5-101	E
5-102	E
5-103	E
5-104	E
5-105	E
5-106	E
5-107	E
5-108	E
5-109	E
5-110	E
5-111	E
5-112	E
5-113	E
5-114	E
5-115	E
5-116	E
5-117	E
5-118	E
5-119	E
5-120	E
5-121	E
5-122	E
5-123	E
5-124	E
5-125	E
5-126	E
5-127	E
5-128	E
5-129	E
5-130	E
5-131	E
5-132	E
5-133	E
5-134	E
5-135	E
5-136	E
5-137	E
5-138	E
5-139	E
5-140	E
5-141	E
5-142	E
5-143	E
5-144	E
5-145	E
5-146	E
5-147	E
5-148	E
5-149	E
5-150	E
5-151	E
5-152	E
5-153	E
5-154	E
5-155	E
5-156	E
5-157	E
5-158	E
5-159	E
5-160	E
5-161	E
5-162	E
5-163	E
5-164	E
5-165	E
5-166	E
5-167	E
5-168	E
5-169	E
5-170	E
5-171	E
5-172	E
5-173	E
5-174	E
5-175	E
5-176	E
5-177	E
5-178	E
5-179	E
5-180	E
Divider	-
A-1	E
A-2	D
A-3	E
A-4	E
A-5	E
A-6	E
A-7	E
A-8	D
A-9	D
A-10	D
A-11	E
A-12	D
A-13	D
A-14	D
A-15	D
B-1	D
B-2	E
B-3	E
B-4	E
B-5	E
B-6	E
B-7	E
B-8	E
B-9	E
B-10	D
B-11	E
B-12	E
B-13	D
B-14	D
B-15	E
B-16	E
B-17	E
B-18	E
B-19	E
B-20	E
B-21	E
B-22	E
B-23	E
C-1	E
C-2	E
C-3	E
C-4	E
C-5	E
C-6	E
C-7	E

PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV	PAGE	REV
C-8	E								
C-9	E								
C-10	E								
C-11	E								
D-1	E								
D-2	E								
D-3	E								
D-4	E								
D-5	E								
D-6	E								
D-7	E								
E-1	E								
E-2	E								
F-1	E								
G-1	E								
G-2	E								
Divider	-								
Index-1	E								
Index-2	E								
Index-3	E								
Index-4	E								
Index-5	E								
Index-6	E								
Comment Sheet	E								
Back Cover	-								

PREFACE

PURPOSE

This handbook describes the installation of the CONTROL DATA® Network Operating System (NOS) Version 2.4.1 and its products. NOS controls the operation of the following Computer Systems:

- CDC® CYBER 180 Computer Systems
  Models 810, 830, 835, 840, 845, 850, 855, and 860

- CDC CYBER 170 Computer Systems
  Models 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 815, 825, 835, 845, 855, 865, and 875

- CDC CYBER 70 Computer Systems
  Models 71, 72, 73, and 74

- CDC 6000 Computer Systems

AUDIENCE

Because there are two methods for installing NOS and its product set, this handbook was written for two types of users: installers of a standard system and installers of a customized system.

If you will be installing a standard system (that is, you will not be modifying the software), you should be familiar with the basic operations of a Control Data computer. For example, you should know how to assign and label tapes and how to enter commands at the system console. For this information, refer to the NOS Operations Handbook and the NOS Reference Set Volume 3.

If you will be customizing the software during installation, you should be familiar with NOS, COMPASS, and each product you are customizing. In other words, you should be familiar with the information provided in the NOS 2 Analysis Handbook and with the information provided in reference manuals for the products you are customizing.

MANUAL ORGANIZATION

This handbook describes the software installation of NOS and its entire product set. You can disregard any information about products you have not ordered.

This handbook is organized in the following manner:

- Section 1 briefly describes standard and customized installations and suggests how to prepare for either one.

- Section 2 describes how to install a standard system using the released deadstart tape. If you do not need to customize NOS or any products, follow the procedure detailed in this section.

- Sections 3 and 4 describe how you can install a customized system by building products from the source code on the permanent file dump tape. Because this method of installation is complex, you should use this method only if necessary. To use this method, you should be familiar with NOS and the products you are installing.

- Section 5 contains special information for customized installation procedures for NOS and each of its products. Before building products, refer to this section for details concerning a specific installation procedure.

- The six appendixes contain reference information: a glossary of terms; a list of installation commands, parameters, and procedures; the file formats of the source materials; controlware installation; special information for 63-character set installation; and Remote Diagnostic Facility information.

CONVENTIONS

The following conventions are used throughout this handbook:

- CYBER 170 and 180 Computer Systems that share functional and architectural attributes are called CYBER 180-class machines. For example, the CYBER 180 Computer Systems and the CYBER 170 Models 815, 825, 835, 845, and 855 are called CYBER 180-class machines.

- The term CYBER 70 Computer Systems refers to models 71, 72, 73, and 74 only.

- The terms command and control statement are interchangeable.

- Extended memory for model 176 is large central memory extended (LCME). Extended memory for the CYBER 180-class machines is unified extended memory (UEM). Extended memory for models 865 and 875 includes unified extended memory and may also include extended core storage (ECS) or extended semiconductor memory (ESM). Extended memory for all other NOS computer systems is either ECS or ESM. ECS and ESM are the only forms of extended memory that can be shared in an ECS multimainframe complex and can be accessed by a distributive data path (DDP).

- ECS refers to both ECS and ESM, and extended memory refers to all forms of extended memory unless otherwise noted. However, when referencing extended memory in the context of an ECS multimainframe complex or DDP access, UEM and LCME are excluded.

- Uppercase letters within command formats must be entered exactly as given; replace lowercase letters with appropriate characters as described in the text.

- At all sites, you must specify Job and USER commands. The CHARGE command is not required at all sites.

- Press CR means press the carriage return key on the CC545 console or press the NEXT key on the CC634B console.

SUBMITTING COMMENTS

The last page of this manual is a comment sheet. Use the comment sheet to suggest specific improvements for the manual and to report any errors. If the comment sheet has already been used, you can mail your comments to:

Control Data Corporation
Publications and Graphics Division
ARH219
4201 Lexington Avenue South
St. Paul, MN 55112

If you have access to SOLVER, an online problem reporting facility, you can use it to submit comments about the manual. Declare your problem type as DOC and use NS2 as the product identifier.

RELATED PUBLICATIONS

Control Data manuals are available through Control Data sales offices or Control Data Literature and Distribution Services (308 North Dale Street, St. Paul, Minnesota 55103).

The NOS Version 2 Manual Abstracts (publication number 60485500) is a pocket-sized manual containing brief descriptions of the contents and intended audience of all NOS 2 manuals and optional product manuals.

Control Data also publishes a Software Publications Release History (publication number 60481000) of all software manuals and revision packets it has issued. This history lists the revision level of a particular manual that corresponds to the level of software installed at the site.

Programming information for the various forms of extended memory can be found in the COMPASS Reference Manual and in the appropriate computer system hardware reference manual.

EXTENDED MEMORY MANUALS

Control Data Publication	Publication Number
Extended Semiconductor Memory Hardware Reference Manual	60455990
Extended Core Storage Reference Manual	60347100
Extended Core Storage II and Distributive Data Path Reference Manual	60430000

NOS 2 MANUALS

Control Data Publication	Publication Number
Common Memory Manager Version 1 Reference Manual	60499200
COMPASS Version 3 Reference Manual	60492600
CYBER Initialization Package (CIP) User's Handbook	60457180
CYBER Loader Version 1 Reference Manual	60429800
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300
CYBER Record Manager Basic Access Methods Version 1 Reference Manual	60495700
CYBER Supermini Operations Handbook	60459850
FORM Version 1 Reference Manual	60496200
Modify Version 1 Reference Manual	60450100
NOS Version 2 Administration Handbook	60459840
NOS Version 2 Analysis Handbook	60459300
NOS Version 2 Applications Programmer's Instant	60459360
NOS Version 2 Diagnostic Index	60459390
NOS Version 2 Network Terminal User's Instant	60459380
NOS Version 2 Operations Handbook	60459310
Nos Version 2 Applications Installation Handbook	84002760
NOS Version 2 Reference Set, Volume 3, Systems Commands	60459680
NOS Version 2 Reference Set, Volume 4, Program Interface	60459690
NOS Version 2 Systems Programmer's Instant	60459370
NOS Version 2 Security Administrator's Handbook	60460410
On-Line Maintenance Software Reference Manual	60454200
SYMPL Version 1 Reference Manual	60496400

OPTIONAL PRODUCT MANUALS

Control Data Publication	Publication Number
ALGOL-60 Version 5 Reference Manual	60481600
APL Version 2 Reference Manual	60454000
BASIC Version 3 Reference Manual	19983900
COBOL Version 5 Reference Manual	60497100
Concurrent Maintenance Library Reference Manual	60455980
Conversion Aids System Version 3 Reference Manual	19265358
CYBER Cross System Version 1 Build Utilities Reference Manual	60471200
CYBER Cross System Version 1 Macro Assembler Reference Manual	96836500
CYBER Cross System Version 1 Micro Assembler Reference Manual	96836400
CYBER Cross System Version 1 PASCAL Reference Manual	96836100
CYBER Interactive Debug Version 1 Reference Manual	60481400
Data Catalogue 2 Version 2 Reference Manual	60483350
DMS-170 CYBER Database Control System Version 2 Data Administration Reference Manual	60485200
DMS-170 CYBER Database Control System Version 2 Application Programming Reference Manual	60485300
DMS-170 Query Update/CYBER Record Manager Data Administration Reference Manual	60482100
FORTRAN Data Base Facility Version 1 Reference Manual	60482200
FORTRAN Extended Version 4 Common Library Mathematical Routines Reference Manual	60498200
FORTRAN Extended Version 4 Reference Manual	60497800
FORTRAN Extended Version 4 to FORTRAN Version 5 Conversion Aids Program Version 1 Reference Manual	60483000
FORTRAN Version 5 Common Library Mathematical Routines Reference Manual	60483100
FORTRAN Version 5 Reference Manual	60481300
MSSI Version 3 Reference Manual	60458820
MSSI Version 3 Installation Handbook	60458830

Control Data Publication	Publication Number
Message Control System Version 1 Reference Manual	60480300
Network Access Method Version 1/Communications Control Program Version 3 Host Application Programming Reference Manual	60499500
Network Access Method Version 1/Communications Control Program Version 3 Terminal Interfaces Reference Manual	60480600
Network Definition Language Version 1 Reference Manual	60480000
NOS Version 2 Applications Installation Handbook	84002760
NOS Full Screen Editor User's Guide	60460420
PASCAL Version 1.1 Reference Manual	60497700
PL/I Version 1 Reference Manual	60388100
Query Update Version 3 Reference Manual	60498300
Remote Batch Facility Version 1 Reference Manual	60499600
Remote Host Facility Access Method Reference Manual	60459990
Screen Formatting Reference Manual	60460430
STIMULA 1 Reference Manual	60234800
Sort/Merge Versions 4 and 1 Instant Manual	60497600
Sort/Merge Versions 4 and 1 Reference Manual	60497500
Sort/Merge Version 5 Reference Manual	60484800
TAF Version 1 Reference Manual	60459500
TAF/CRM Data Manager Version 1 Reference Manual	60459510
Text Editor Version 1 Reference Manual	60436100
Update Version 1 Reference Manual	60449900
XEDIT Version 3 Reference Manual	60455730
5870 Printer Site Reference Manual	60462710
8-Bit Subroutines Version 1 Reference Manual	60495500

HARDWARE MANUALS

Control Data Publication	Publication Number
CYBER 70 Computer System Model 71 Hardware Reference Manual	60453300
CYBER 70 Computer System Model 72 Hardware Reference Manual	60347000
CYBER 70 Computer System Model 73 Hardware Reference Manual	60347200
CYBER 70 Computer System Model 74 Hardware Reference Manual	60347400
CYBER 170 Computer System Hardware Reference Manual	60420000
CYBER 170 Computer Systems Models 720, 730, 740, 750, 760, and 176 Level B/C Hardware Reference Manual	60456100
CYBER 170 Computer System Models 815 and 825 Hardware Reference Manual	60469350
CYBER 170 Computer Systems Models 865 and 875 Hardware Reference Manual	60458920
CYBER 170/180 Computer Systems Models 835, 845, and 855 Hardware Operator's Guide	60458390
CYBER 170/180 Computer Systems Models 835, 840, 845, 850, 855, 860, and 990 (CYBER 170-State) Hardware Reference Manual	60469290
CYBER 180 Computer Systems Models 810 and 830 (CYBER 170-State) Hardware Reference Manual	60469420
CYBER 180 Computer Systems Models 810 and 830 Hardware Operator's Guide	60469440
380-170 Network Access Device Hardware Reference Manual	60458500
6400/6500/6600 Computer Systems Reference Manual	60100000
2550-2 Host Communication Processor Hardware Reference Manual	74375500

DISCLAIMER

NOS and its product set are intended to be used only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

CONTENTS

APPENDIXES

INDEX

FIGURES

TABLES

INTRODUCTION 1

WHAT IS INSTALLATION?

Installation is the process of creating an operating system deadstart file and then loading the system. Loading the system means copying the information on the deadstart file to specific locations in central memory and to disk. The deadstart file, which can reside on tape or disk, contains binary information created by assembling operating system and product set source code.

In addition to the information on the deadstart file, you must also copy several permanent files to disk. Most of the permanent files you need are released on the permanent file tapes. You create the rest of the permanent files by executing various installation procedures.

STANDARD AND CUSTOMIZED INSTALLATIONS

CDC releases NOS and all the products you have ordered on the deadstart tape. This tape has been tailored according to the information you provided in the Order Information Package (OIP).

Using the released deadstart tape to install the system is a standard installation. Standard installation requires a minimum number of steps. By using it, you do not have to build products from source code, nor do you have to verify the installation; CDC has already performed those procedures for you. Section 2 tells you how to install a standard system.

If you need to modify NOS or any products, you can do so by building products from source code. This is a customized installation. A customized installation is complex and requires many procedures; it also requires more time to perform than a standard installation. Note, however, that all products are prebuilt by CDC and released on the deadstart tape, so you need only build the products you want to customize. Sections 3, 4, and 5 describe how to install a customized system.

PREPARING FOR AN INSTALLATION

To perform either a standard or customized installation you need to:

- Ensure that CIP has been installed to disk.
- Collect all release tapes.
- Collect and update manuals.
- Read all release bulletins.

These tasks are described in the following paragraphs.

ENSURE THAT CIP HAS BEEN INSTALLED TO DISK

The CYBER Initialization Package (CIP) tape contains hardware and software interface modules that allow NOS to run on a CYBER mainframe. CIP is released separately from your NOS order and is sent to your customer engineer (CE). If you have an 800 series computer, the CIP modules must be installed to disk. If you have a non-800 series computer, installing the CIP modules is optional. Check with your CE to ensure that CIP has been installed; CIP installation should be a cooperative effort between you and your CE.

COLLECT ALL RELEASE TAPES

You need these tapes to install NOS and any optional products:

- Deadstart Tape

  This tape contains all the executable programs for NOS and the products you have ordered. These programs were built according to the information you specified in the OIP.

- Permanent File Tapes

  The permanent file tapes contain the permanent files your system needs to run. (These tapes also contain the source code for your system. You can use the source code if you want to perform a customized installation.)

NOTE

If you ordered only a software component (and not NOS), the executable programs and source code for your component order are released on permanent file tapes.

COLLECT AND UPDATE MANUALS

A complement of manuals is sent with each release. Manuals that have been completely revised since the last edition have a yellow cover sheet. You can discard your last edition of these manuals. Revision packets, if any, have a pink cover sheet. The pink cover sheet explains which pages are to be inserted or removed from the last edition of the manual.

In addition to using this handbook, you may need to refer to the following publications while performing an installation:

Control Data Publication	Publication Number
NOS Version 2 Administration Handbook	60459840
NOS Version 2 Analysis Handbook	60459300
CYBER Initialization Package (CIP) User's Handbook	60457180
NOS Version 2 Reference Set, Volume 3, System Commands	60459680
NOS Version 2 Operations Handbook	60459310
NOS Version 2 Systems Programmer's Instant	60459370

READ ALL RELEASE BULLETINS

There are two types of documents you should read before performing an installation: the Software Release Bulletin and any Installation Bulletins.

A Software Release Bulletin (SRB) is issued with each NOS release. This bulletin, which is issued after this handbook, contains up-to-the-minute information that is specific to the release.

A printed copy of the SRB is included with your release materials. In addition, a file copy of the SRB is released on the permanent file tape(s). The file (named SRB) is loaded to the installaton user name INSTALL by X.SYSGEN(FULL).

Installation Bulletins (IBs) may be issued to communicate information that is discovered after the release has been forwarded to customers. You can access IBs through SOLVER.

SOLVER is an interactive program that allows you to access the Programming Systems Report (PSR) database. By using SOLVER, you can:

- List IBs associated with the system you are installing.
- Report software problems to CDC.
- Check the status of problem reports.
- Search for duplicate problem reports submitted by other customers.

You can access SOLVER through direct dial or Telenet lines. Contact your local CDC Professional Services office for complete access information.

INSTALLING A STANDARD SYSTEM 2

This section describes how to install a standard (unmodified) system for both a full product order and for a component order.

A full product order is an order that includes NOS. If your order includes NOS, you receive a deadstart tape and permanent file tapes. To install a full product order, follow the steps in Full Product Order Installation.

After ordering and installing this release of NOS, you may decide to order additional products. This is called a component order. For a component order, you receive only the permanent file tapes. You need to use your deadstart file from the previous full product order of the same PSR level. To install a component order, follow the steps in Component Order Installation.

FULL PRODUCT ORDER INSTALLATION

A full product order installation consists of the following four steps, which are described in detail in this section.

Step 1: Deadstart your system using the released deadstart tape.

Step 2: Install the permanent files necessary for running NOS.

Step 3: Create and move necessary configuration files.

Step 4: Create a new deadstart file.

STEP 1: DEADSTART USING THE RELEASED DEADSTART TAPE

During the deadstart process, you must define your hardware configuration and other system attributes by making entries to the deadstart decks on the deadstart tape. CDC provides you with a set of preconfigured deadstart decks. These preconfigured decks are listed in the Software Release Bulletin (SRB).

When you deadstart the system, use the deadstart decks that most closely match your particular configuration. Once you have deadstarted the system using the preconfigured decks, you must then modify the decks for your particular hardware configuration. For example, you need to make entries to specify the number and type of disk and tape units connected to your computer. Here is a brief description of each deck.

- CMRDECK (Central Memory Resident Deck). A CMRDECK is a text record on the deadstart tape. The entries in this deck describe the central memory table sizes to be used by the system; this deck also specifies which EQPDECK, IPRDECK, and LIBDECK is used by the system.

- EQPDECK (Equipment Deck). An EQPDECK is a text record on the deadstart tape. The entries in this deck describe the hardware connected to your computer. These entries are used by the system to build the equipment status table (EST).

- IPRDECK (Installation Parameters Deck). An IPRDECK is a text record on the deadstart tape. The entries in this deck describe the default mode of system operation.

- LIBDECK (Library Edit Deck). A LIBDECK is a SYSEDIT directive record on the deadstart tape. The entries in this deck specify program residence, field length, record type, and parameter format. This deck cannot be modified during the deadstart process.

Write down any deadstart deck entries you make during the deadstart process. You will need this information in Step 4, Creating a New Deadstart File.

The NOS 2 Operations Handbook contains detailed explanations of the deadstart process; the NOS 2 Analysis Handbook contains detailed descriptions of the deadstart decks.

To deadstart your system, follow these steps:

1. Mount the deadstart tape on an available tape unit.

2. Set your deadstart panel (or deadstart display) to specify a level 0 deadstart from the preconfigured CMRDECK you want to use to deadstart your system. (For detailed information about setting the deadstart panel, refer to the NOS 2 Operations Handbook.)

   - For CYBER 800 series computers, set up the panel for a disk deadstart from the disk unit that contains CYBER Initialization Package (CIP).

   - For non-800 series computers (where no CIP has been installed), specify a tape deadstart from the unit where you mounted the deadstart tape.

3. Initiate the deadstart sequence.

   - If you are using the CC545 console, press the deadstart button.

   - If you are using a CC634B console, press the RESET key to reinitialize the console. Next, press CTRL G and then press CTRL R.

   - For models 810, 815, 825, and 830, press CR to advance to the Initial Options display.

   - For all other models, the Initial Options display appears after you initiate the deadstart sequence.

4. When the Initial Options display appears, do one of the following:

   - For all CYBER 800 series computers, enter O to select operator intervention and then enter T to specify operating system deadstart from tape. Next, enter the tape channel, equipment, and unit numbers that describe where the deadstart tape is mounted.

   - For all other models, press the CR key.

5. Enter CMRDECK entries.

   - An instructional display (CMRINST) appears that gives the format for CMRDECK entries. Do the following to page to the screen where you enter data:

     - On the CC545 console, press the right blank key.

     - On the CC634B console, press the tab key.

   - Enter any CMRDECK changes. When you have finished, enter:

     NEXT.

6. Enter EQPDECK entries.

   - An instructional display (EQPINST) appears that gives the format for EQPDECK entries. Do the following to page to the screen where you enter data:

     - On the CC545 console, press the right blank key.

     - On the CC634B console, press the tab key.

   - Make additional EQPDECK entries to describe your hardware configuration. Use the recommended CYBER channel configuration conventions. These channel conventions are listed in appendix G.

   - When you have finished making the EQPDECK entries, enter:

     GO.

The system now begins loading. If you are requested to do so, enter the current date and time. When the system is loaded, the initial A and B displays appear.

STEP 2: INSTALL PERMANENT FILES

The permanent files for your system are released on the permanent file tapes. You install these files by using the SYSGEN command. (For a detailed explanation of SYSGEN and all SYSGEN options, refer to appendix B.)

Before using SYSGEN to install permanent files, note the following:

- SYSGEN User Names and Passwords

  The SYSGEN command uses certain user names and passwords to install the necessary permanent files. (User names and passwords are stored in a validation file.) If you do not have a validation file, SYSGEN creates a file with the validations necessary for installation. If you have a validation file, SYSGEN does not modify the file. If you want to use your own validation file, you must temporarily modify the file so that SYSGEN can find the user names and passwords it needs, or you can modify SYSGEN. Refer to appendix B for more information.

- Permanent File Replacement

  If you have permanent files on your disk with the same names as the released permanent files, SYSGEN replaces these files with the released version. If you do not want these files replaced, make a backup copy of the files you want to save before using the SYSGEN command. Then, after using SYSGEN to install permanent files, replace the files on disk with your backup copies of the files. For a list of the permanent files that are automatically installed by SYSGEN, refer to appendix B.

- Installation of Online Manuals

  The permanent file tapes contain both the source and bound files for each online manual. SYSGEN automatically installs all the bound online manual files and a procedure file called MANLOAD on user name MANUALS. The bound version of the online manuals is used by the EXPLAIN command. If you want to modify any of the online manuals, you can use the MANLOAD procedure to install the source version of the manuals. Also, because the online manuals use a great amount of disk space, you should delete any manuals you do not need once installation is completed.

- Installation of Concurrent Maintenance Library (CML).

  The permanent file tapes contain the most recent version of CML at the time NOS released. It is automatically installed to user name CDCCE by X.SYSGEN(FULL). You should inform your customer engineer (CE) that the file is available. Also, because the file takes up a large amount of disk space, you should request that your CE remove any unnecessary diagnostics once installaion is complete.

To install all permanent files, follow these steps:

1. Enter the following command:

   ```
   X.SYSGEN(FULL)
   ```

2. Mount the requested permanent file tape on an appropriate tape unit when you are prompted to do so. The volume serial number (VSN) for each tape is printed in the media number field of the external tape label. SYSGEN automatically assigns the tape.

When this procedure is completed, all the necessary permanent files are set up on your default family device.

STEP 3: CREATE AND MOVE CONFIGURATION FILES

There are four types of configuration files you might need to create:

- A Logical Identifier Configuration File
- A Network Configuration File and a Local Configuration File
- A Remote Host Facility Configuration File
- A Mass Storage Archival Subsystem Configuration File

Logical Identifier Configuration File

If you ordered the NOS Scope Station, the Remote Host Facility (RHF), or the Permanent and Queued File Transfer Facility (PTF/QTF), you must create the Logical Identifier Configuration File (LIDCMid) to define the physical and logical machines available to your system. The system uses this file to automatically create a Logical Identifier Table (LDT) in central memory. Refer to the NOS 2 Analysis Handbook for information about creating an LIDCMid file.

Network and Local Configuration Files

If you installed NAM, you must create a Network Configuration File (NCF) and a Local Configuration File (LCF). To do so, use the Network Definition Language (NDL). A sample network configuration file called NDLDATA is automatically installed by SYSGEN(FULL) on the user name INSTALL.

SYSGEN automatically created a Communications Control Program (CCP) load file which is referenced by the NCF. Refer to the SRB for the list of Network Processing Unit (NPU) variants you received. If the released variants listed in the SRB are not acceptable, or if you want to add CCP online diagnostics or the 3270 Terminal Interface Program (TIP) to your variants, you must create your own variant from source code.

To help you build CCP and Cross, a partially built CCP is supplied on the permanent file tapes. Refer to section 5 for more information about CCP and Cross installation.

Refer to section 5 and the NDL Reference Manual for more information about NCFs.

Remote Host Facility Configuration File

If you installed the RHF, you must create an RHF Configuration File (RCFMid). Refer to the NOS 2 Analysis Handbook for more information.

Mass Storage Archival Subsystem Configuration File

If you installed the Mass Storage Archival Subsystem (MAS), you must create a MAS configuration file. Refer to the NOS 2 Analysis Handbook for more information.

STEP 4: CREATE A NEW DEADSTART FILE

To create a new deadstart tape that contains entries for your site, follow these steps:

1. Get a copy of the deadstart decks (CMRDECK, EQPDECK, IPRDECK, LIBDECK, and APRDECK) you used in Step 1, Deadstart Using the Released Deadstart Tape. Use the following commands:

   ```
   X.DIS.
   COMMON(SYSTEM)
   GTR(SYSTEM,LGO)deckname1,deckname2,...,decknamen
   ```

   Replace $deckname_1$ through $deckname_n$ with the names of the deadstart decks you want to update. (For example, CMRD03 for CMRDECK 03, EQPD03 for EQPDECK 03, and so forth).

2. Use an available editor to make any necessary changes to the deadstart decks on file LGO. On 800 series computers, you can use the Remote Diagnostic Facility (RDF) as a means to using some NOS editors. Refer to appendix F.

3. When you have made all the necessary changes to the deadstart decks, use the following commands to merge the records on file LGO with the contents of your current system:

   ```
   SYSGEN(DST,SYSTEM,LGO,NEW,0,dd)
   DROP.
   ```

   Replace dd with the tape density you want to use for the new deadstart tape (HY, HD, PE, GE).

4. When you are prompted to mount a tape, mount an unlabeled scratch tape. Then enter this command:

   ```
   VSN,est,NDT.
   ```

   Replace est with the Equipment Status Table (EST) ordinal number of the tape drive unit where your scratch tape is mounted.

When this procedure has finished executing, you have a new deadstart tape to use for subsequent deadstarts.

COMPONENT ORDER INSTALLATION

This section describes how to install a component order using the binaries and source code released on the permanent file tapes. If you plan to customize your component order (build the components from source code), you must execute these commands from your installation name to update the RECLAIM database. Then refer to sections 3 through 5 for information about customizing products.

Before using the procedures detailed in this section, note the following:

- To install a component order, you need the released permanent file tapes and your deadstart file from a previous full product order of the same PSR level. Your validation file must contain the user names and passwords listed in appendix B. You must also perform the component installation from the installation user name INSTALL using this USER command: USER,INSTALL,INSTALL.

- SYSGEN(UPGRADE) merges the binaries on the permanent file tapes with your existing system, automatically updates existing binaries and adds new ones, and optionally writes a new deadstart tape. After you execute the SYSGEN(UPGRADE) command, you can execute the SYSGEN(FILES,vsn) command to install the permanent files. However, if no permanent files need to be installed, the SYSGEN(FILES,vsn) command does nothing.

  There are three ways you can use the SYSGEN(UPGRADE) command:

  - Update the RECLAIM database without creating a deadstart tape.

  - Build a deadstart tape by requesting tapes.

  - Build a deadstart file using local files.

  These options are detailed on the following pages.

UPDATING THE RECLAIM DATABASE

If your component order consists of products which do not add binaries to the deadstart tape, or if you want to update the RECLAIM database without creating a new deadstart file, perform this step:

- Enter this command:

  ```
  SYSGEN(UPGRADE,0,0,vsn,dd)
  ```

  Replace vsn with the VSN of the CDC-supplied permanent file tape. The VSN is listed in the media number field of the external tape label.

  Replace dd with the tape density of the permanent file dump tape (HY, HD, PE, GE).

CREATING A DEADSTART TAPE USING TAPES

To create a new deadstart tape using a previous deadstart tape, follow these steps:

1. Enter this command:

   SYSGEN(UPGRADE,OLD,NEW,vsn,dl,d2)

   Replace vsn with the VSN of the CDC-supplied permanent file tape. The VSN is in the media number field of the tape label.

   Replace dl with the tape density of the permanent file tape (HY, HD, PE, or GE).

   Replace d2 with the tape density of the deadstart tape and the tape density for the new deadstart tape.

2. Mount the current or old deadstart tape with a VSN of ODT (old deadstart tape) when you are prompted to do so (VSN,est,ODT).

3. Mount an unlabeled tape for the new deadstart tape with a VSN of NDT (new deadstart tape) when you are prompted to do so (VSN,est,NDT).

4. Load permanent files from the permanent file tape by entering the following command at the system console:

   X.SYSGEN(FILES,vsn)

   Replace vsn with the VSN you used in step 1.

CREATING A NEW DEADSTART FILE USING LOCAL FILES

This option merges the component-order binaries with your base deadstart file (specified by the old parameter) and produces a new deadstart file (specified by the new parameter).

1. Enter this command:

   SYSGEN(UPGRADE,old,new,vsn,dd)

   Replace old with the file name of the deadstart file you have preassigned.

   Replace new with the file name you want for the new deadstart file. You can preassign this file as a magnetic tape file, direct access file, or a local file. A tape will not be requested for writing the new deadstart file.

   Replace vsn with the VSN of the CDC-supplied permanent file tape. The VSN is in the media number field of the external tape label.

   Replace dd with the tape density of the permanent file tape (HY, HD, PE, or GE).

2. Load permanent files from the permanent file tapes by entering this command at the system console:

   X.SYSGEN(FILES,vsn)

   Replace vsn with the VSN you used in step 1.

PREPARING FOR A CUSTOMIZED INSTALLATION 3

This chapter gives step-by-step instructions for preparing your system for a customized installation. Preparation consists of the following steps:

Step 1: Setting up your permanent file environment.

Step 2: Customizing installation procedures.

Step 3: Creating USER files for input to installation procedures.

Step 4: Creating files GLOBLIB and PRODUCT.

Step 5: Executing the MODOPL procedure to modify NOS installation parameters.

Before you begin preparing to install a customized system, note the following:

- Because all products are prebuilt by CDC, you need only rebuild the products you want to customize, and products that are dependent on the products you are customizing.

- Make a list of the products you want to customize. Next, check the product dependency chart in section 4 (figure 4-1) to find out if other products are dependent on the products you want to customize and add those to the list. These are the products you must rebuild.

- Carefully review your plans for customizing the operating system and products. Refer to section 5 for special information about each product you are installing.

- Special installation parameters, procedures, and commands are detailed in appendix B.

- Special information about installing a 63-character set system is detailed in appendix E. Read this appendix before you begin installing your system.

The following special information is for a customized component order installation:

- If you are installing a customized component order, you must use a deadstart tape from a previous order of the same PSR level.

- Before installing the customized component order, you must first use the SYSGEN(UPGRADE) command to update the RECLAIM database. Follow the steps in Component Order Installation in section 2 of this manual.

- If your component order contains a Modify product that resides on the composite OPL, you must update your existing composite OPL before building the component order product. To do this, first run the SYSGEN(UPGRADE) command as described in section 2. Then use the SYSGEN(RECLAIM) command to load file OPL630 from the component order permanent file tape. This file must be LIBEDITed against your previous composite OPL. (If you have updated the composite OPL with local code, be sure to use your OPL as a basis for the LIBEDIT). Once you have created the merged composite OPL, you can begin your customized installation of the component order.

STEP 1: SETTING UP YOUR PERMANENT FILE ENVIRONMENT

If you are installing on a dedicated system (that is, you have set aside time on your computer to be used only for installation), you can deadstart your system using the released deadstart tape and then use SYSGEN to load all the necessary permanent files. Follow the steps described in Dedicated System Environment Installation.

If you are installing on a production system (that is, you will not be scheduling dedicated time for your installation), you must extract certain files from the release tapes. Follow the steps described in Production Environment Installation.

You must decide where you want to store source files during the installation process. You can either let the installation decks load files automatically from the permanent file tapes or you can preload the files to disk. If you let the installation decks automatically load the required permanent files, you can make the files disk-resident or you can use local files during the installation procedures. Note that some source files must be disk-resident; for example, the composite OPL and numerous CCP build files. Once you have loaded the source installation tools, you can set up files for disk installation. Refer to appendix B for information about controlling the permanent file environment.

DEDICATED SYSTEM ENVIRONMENT INSTALLATION

If you are installing on a dedicated system, follow these steps to prepare your system for a customized installation:

1. Deadstart your system using the deadstart tape. Then, create or modify your deadstart decks (CMRDECK, EQPDECK, IPRDECK, APRDECK, LIBDECK) as necessary.

2. Modify your validation file so that you can use the SYSGEN command to load the necessary permanent files. Refer to SYSGEN validations in appendix B. (Note that SYSGEN(FULL) creates validations for you if you have none.)

3. - If you already have the permanent files necessary to run your system and only need to load the installation procedures, execute the following command from the system console:

     ```
     X.SYSGEN(SOURCE,CCP)
     ```

     Include the keyword CCP only if you want to have CCP/CROSS binary files loaded.

   - If you want to have SYSGEN create all system permanent files in addition to loading installation tools, enter these commands:

     ```
     X.SYSGEN(FULL)
     X.SYSGEN(SOURCE,CCP)
     ```

   Include the keyword CCP only if you want to have CCP/CROSS binary install files loaded.

   Refer to section 5 for more information about the CCP binary installation.

PRODUCTION ENVIRONMENT INSTALLATION

If you are installing on a production environment system, do the following to prepare for a customized installation:

- Mount the released deadstart tape on an available tape unit and enter these commands:

```
LABEL,TAPE,VSN=vsn,D=density,LB=KU,F=I.
COPYEI,TAPE,SYSTEM,V.
RETURN,TAPE.
GTR(SYSTEM,SYSGEN)PROC/SYSGEN
GTR(SYSTEM,PFG)ULIB/PFGLIB
SYSGEN(SOURCE,CCP)
```

  Include the keyword CCP only if you want to load CCP/CROSS binary installation files. Refer to section 5 for more information.

These commands load the RECLAIM database (RECLDB), DECKOPL, INSTALL, CODEPL, MISCPL, NOS PSR report file (PSRRPT), USERBPS, NAMSTRT (if appropriate), and COMMOD and make RECLAIM a permanent file.

In addition to setting up RECLAIM and the RECLAIM database, these commands load numerous other files needed for a source installation. Refer to appendix B for a complete list of files.

STEP 2: CUSTOMIZING THE INSTALLATION PROCEDURES

You build products by executing installation procedures on the INSTALL procedure file. To customize the installation procedures on the INSTALL file, follow these steps:

NOTE

If you want to install a 63-character set, read appendix E before continuing your installation.

1. List the contents of DECKOPL by using the DECKLIS procedure (for details about DECKLIS, refer to appendix B):

   ```
   BEGIN,DECKLIS,INSTALL.
   ```

2. Edit file COMMOD using an available editor. File COMMOD is a correction set containing parameters which control the following:

   - A disk installation.
   - A disk installation using auxiliary packs.
   - A USERCHG file for installation procedures.
   - Different values for other common parameters.

   Refer to COMMOD File Parameters in appendix B for a list of the parameters you can change.

3. Locate the parameters you want to change, and specify the values you want for the parameters. When you have finished, replace file COMMOD.

4. Recreate the INSTALL file to include the new parameter values by entering this command:

   ```
   BEGIN,SETUP,INSTALL,MOD=COMMOD,INSTALL.
   ```

STEP 3: CREATING USER FILES

To further customize products during installation, you can create USER files (files containing modified code for each product you want to customize). Each file can contain the following:

- Programming Systems Report (PSR) corrective code.
- MISCPL code.
- Product installation parameter settings (for specific parameters for a product, refer to the product entry in section 5).
- Site code.

Refer to section 5 and the Software Release Bulletin (SRB) for special information about each product you are installing.

When you create a USER file that contains modifications for a product, these modifications must be in the same format as the product you are modifying, either Modify or Update. NOS, NOS products, and APL 2 are in Modify format. All other products are in Update format.

USERF PARAMETER

When an installation job is executing, it expects to find your modifications on a local file named USER. However, if you do not want to make your modification file local to the installation job, you can use the USERF parameter on the installation procedure call to specify the permanent file that contains your modifications. For example, you can store local modifications for the product AAM2 on the permanent file USERAAM. Then, when you are calling the AAM2 installation procedure to build AAM2, you can use the USERF parameter on the procedure call to pass the file name of the modification file USERAAM.

```
BEGIN,AAM2,INSTALL,USERF=USERAAM.
```

NOTE

If you use file USER to add site modifications, the code is not installed on the program library. This code is applied last and is added to the generated binaries. If you need this program library as input for another product installation, all the code is not present and problems could result. To avoid this, create a new program library containing your user code.

STEP 4: CREATING FILES GLOBLIB AND PRODUCT

The SEED procedure creates three files: PRODUCT, DIRFILE, and GLOBLIB. The binaries in PRODUCT and GLOBLIB are used to satisfy all dependencies for the start of a full installation, as well as dependencies for a partial installation. DIRFILE is used to keep track of all the binaries in file PRODUCT.

SEED places necessary user libraries on file PRODUCT and places the released binaries for COMPASS, SYMPL, UPDATE, COPYL, ITEMIZE, LOADER, TDUEX, PASCAL, and procedures PDU and ULIB on file GLOBLIB. SEED also places the following text binaries on GLOBLIB: LDRTEXT, NETTEXT, SMTEXT, NOSTEXT, PSSTEXT, SYSTEXT, CTITEXT, IOTEXT, SSYTEXT, PPTEXT, and CETEXT. To aid in the installation process, the binaries for the released versions of RECLAIM, NDLP, and SYSGEN are placed in file GLOBLIB. The installation jobs use these binaries instead of those in the running system.

To execute the SEED procedure, you must have a copy of the released system. If you have deadstarted using the released deadstart tape, you can enter COMMON(SYSTEM). If not, issue a tape LABEL or REQUEST command for the released deadstart tape.

Here is the format for the SEED call:

```
BEGIN,SEED,INSTALL,param1,param2,....
```

Parameter	Description
DST=dstname	Local file name for your copy of the system. If the file dstname is not a mass storage file, dstname is copied to a mass storage file before the GTR command in the SEED procedure extracts the required binaries.
REBUILD	The keyword parameter, REBUILD, causes SEED to load binaries into GLOBLIB and PRODUCT to satisfy all dependencies. You should specify REBUILD if you will not be starting your installation procedures with the first product on the dependency chart.
RESET	The keyword parameter RESET causes SEED to initialize the files JOBSTAT and DAYFILES; these files keep statistics on the installation procedure jobs and dayfiles. To get a report of the resource usage of your installation jobs and a list of what jobs have passed or failed, refer to REPORT Procedure in appendix B.

STEP 5: EXECUTING THE MODOPL PROCEDURE TO MODIFY NOS INSTALLATION PARAMETERS

Use the MODOPL procedure to apply code (for example, installation parameters) against the composite OPL. (The OPL contains all Modify-formatted products except APL and TDU.)

If you are not applying any modifications to the composite OPL, you do not need to execute MODOPL. The OPL will be automatically loaded from tape when it is needed. If you execute MODOPL, it creates a disk file OPLpsrout (psrout defaults to the current release level; for example, OPL630). If DISKINS=NO, MODOPL also writes a copy to tape.

MODOPL always writes the updated program library to a permanent file named OPLpsrout. The common deck COMSOPL defines its residency. OPLpsrout is used as an auxiliary PL by many installation jobs. The operating system installation jobs use it to generate the compile file for operating system installations.

For execution of the MODOPL procedure, create file USER to include the following items:

- Miscellaneous code (optional) from MISCPL.
- Modifications of the NOS installation parameters (refer to section 5).
- Modifications to operating system products (refer to section 5).
- Site code.

To execute the MODOPL procedure, enter this command:

```
BEGIN,MODOPL,INSTALL.
```

THE CUSTOMIZED INSTALLATION PROCESS 4

After you have prepared your system for a customized installation by following the steps in the preceding section, you are ready to begin the actual installation process, which consists of these four steps:

Step 1: Building products from source code.

Step 2: Moving permanent files to appropriate user names.

Step 3: Creating a new deadstart tape.

Step 4: Verifying your installation.

These steps are detailed on the following pages.

STEP 1: BUILDING PRODUCTS FROM SOURCE CODE

RUNNING INSTALLATION PROCEDURES

You build a product from source code by executing an installation procedure from file INSTALL. You can call the procedures interactively or you can submit them as batch jobs. If you call a procedure interactively, the procedure submits a batch job for execution (unless you have set IA=YES; refer to COMMOD file parameters in appendix B). If you require file USER for an interactively called procedure, you must specify the USERF=pfn parameter (refer to USERF Parameter in section 3) on the call to the procedure. Here is the format for calling an installation procedure:

```
BEGIN,procname,p1,p2,p3,...,pn.
```

Replace procname with the installation procedure name. These names are listed in tables 4-1, 4-2, 4-3, 4-4, 4-5, and 4-6. The optional parameters (p1 through pn) include two types of parameters: common and unique. Refer to COMMOD file parameters in appendix B for a list of common parameters; refer to section 5 for the unique parameters of a specific installation procedure.

If you are installing a customized component order, refer to figure 4-1 and table 4-7. Figure 4-1 shows the product dependencies and table 4-7 lists the OIP products and the DECKOPL build procedures that you need for assembling each product.

PRODUCT DEPENDENCIES

Some installation procedures require the output of another installation procedure. These procedures are dependent on other procedures; that is, they cannot be run until the procedure they depend on is completed. Figure 4-1 illustrates the product installation procedure dependencies.

Based on whether a procedure is dependent on another, the procedures have been separated into six groups. Run the procedures in the order described below.

Run Group 1

Run the group 1 procedures in the order in which they are listed in table 4-1. Do not run a procedure until the preceding procedure has finished.

Run Group 2

Run the group 2 procedures listed in table 4-2 after running all the procedures in group 1. You can run them in any order you want.

Run Group 3

Run the group 3 procedures listed in table 4-3 after all the procedures in group 1. Group 3 procedures must be run in the order listed in table 4-3.

Run Group 4

Run the group 4 procedures listed in table 4-4 after all the procedures in group 1. Group 4 procedures must be run in the order listed in table 4-4.

Run Group 5

Run the group 5 procedures listed in table 4-5 after all the procedures in group 1 and after the AAM2 procedure from group 3. Group 5 procedures must be run in the order listed in table 4-5.

Run Group 6

Run the NOS2B procedure after all the group 1 procedures, the FTN5 procedure from group 4, and the NAM5 procedure from group 5.

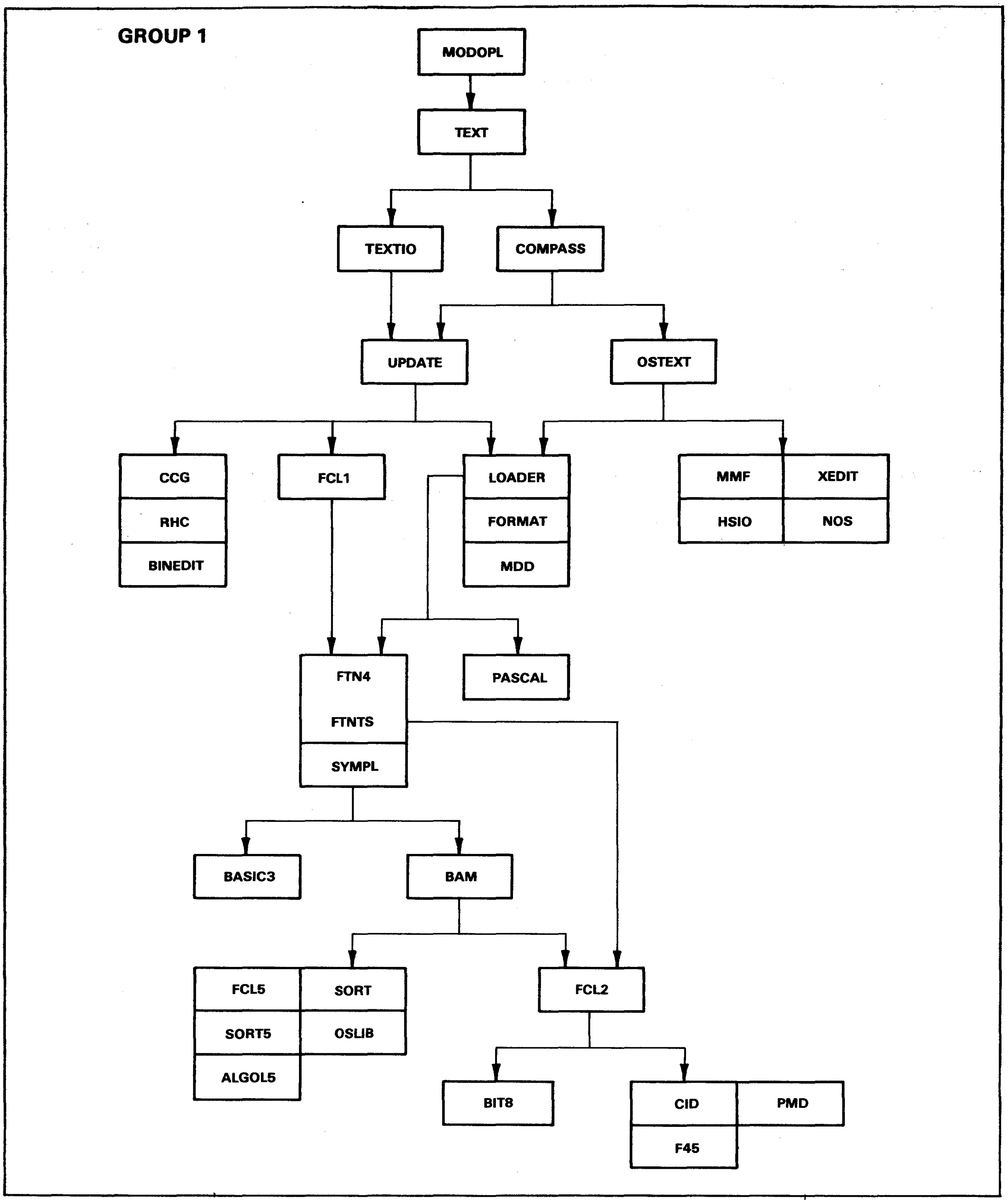

Figure 4-1. Build Dependencies Chart (Sheet 1 of 3)

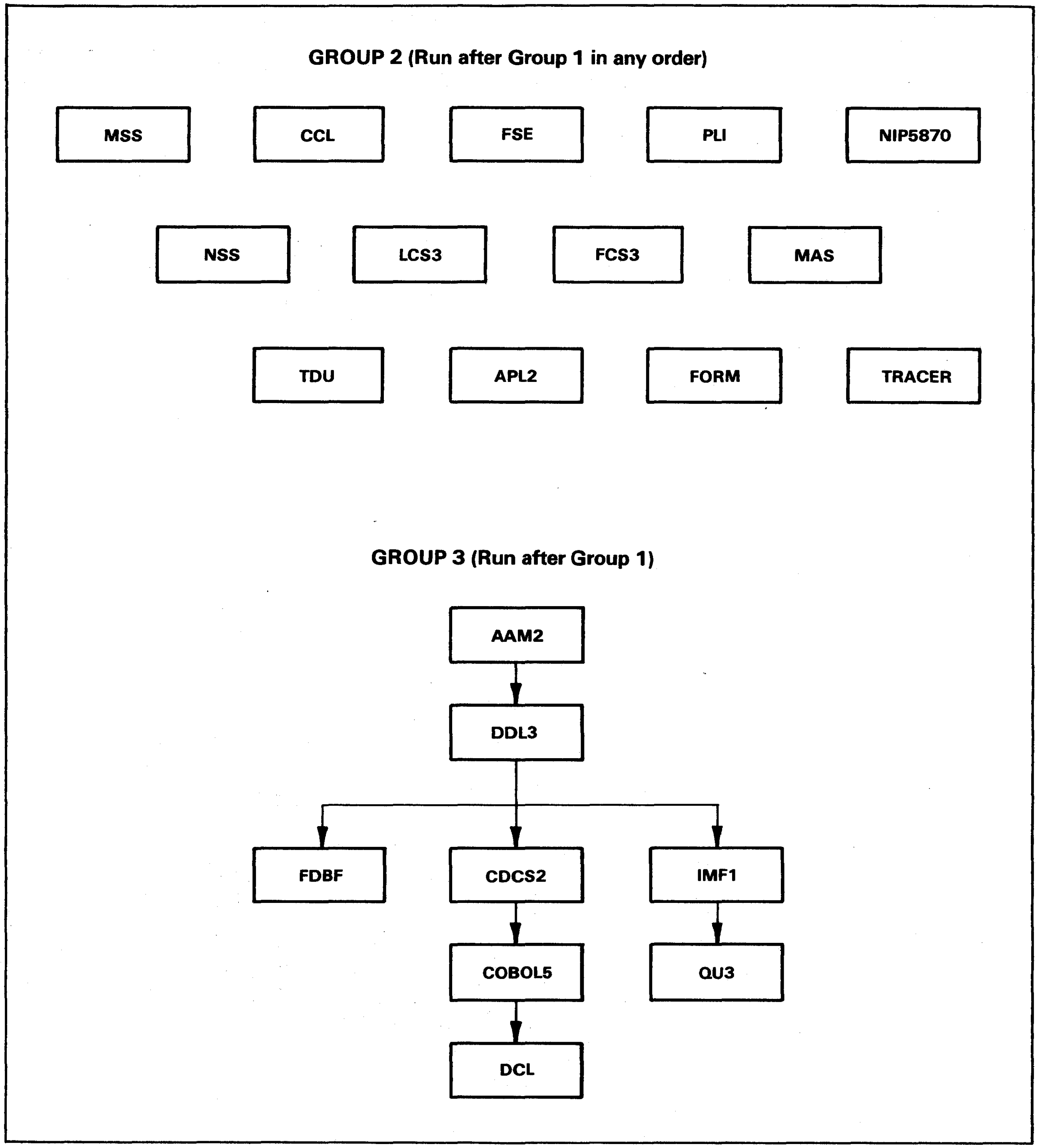

Figure 4-1. Build Dependencies Chart (Sheet 2 of 3)

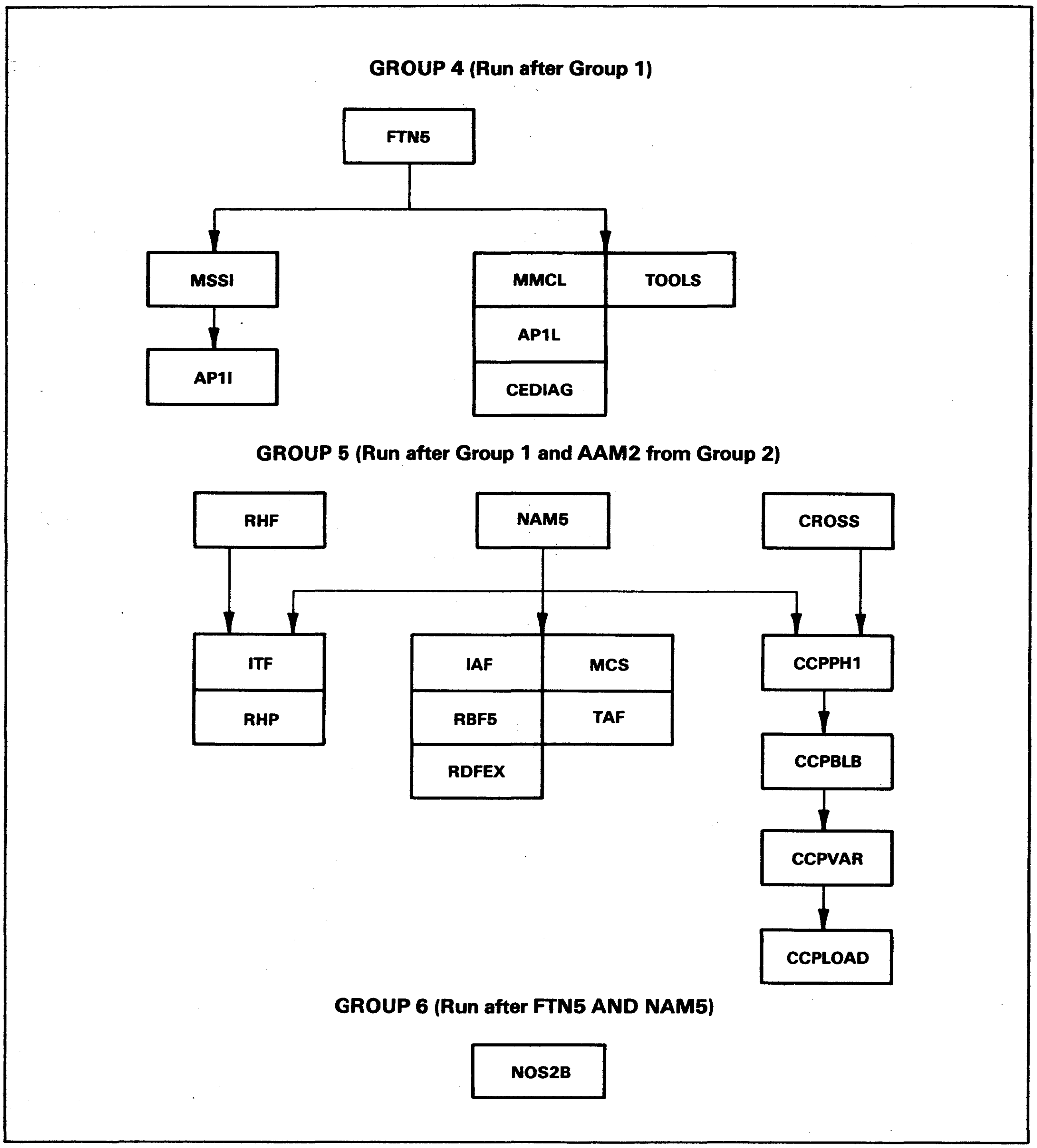

Figure 4-1. Build Dependencies Chart (Sheet 3 of 3)

Table 4-1. Group 1 Products
(Sheet 1 of 2)

Procedure Name	Unique Keywords	Required Files
MODOPL		OPLpsrin
TEXT	MFT=mft ECS=ecs CSET=cset	TEXTpsrin
TEXTIO		TXIOpsrin
COMPASS		CPS1psrin
UPDATE		UPD1psrin CPS1psrout
OSTEXT		OPLpsrout
CCG		CCG1psrin
RHC		RHC1psrin
BINEDIT		BINEpsrin CPS1psrout
FCL1		FCL4psrin CPS1psrout
LOADER	PRESET=value MAP=map FLMSG=flmsg	LDR1psrin
FORMAT		FMATpsrin OPLpsrout
MDD		MDD1psrin OPLpsrout
MMF		OPLpsrout
HSIO		OPLpsrout
XEDIT		OPLpsrout
NOS		OPLpsrout
FTN4		FTN4psrin CPS1psrout

Table 4-1. Group 1 Products
(Sheet 2 of 2)

Procedure Name	Unique Keywords	Required Files
FTNTS		FTI4psrin CPS1psrout
SYMPL		SYMPpsrin
PASCAL		PASCpsrin OPLpsrout
BASIC3		BAS3psrin CPS1psrout OPLpsrout
BAM		BAM1psrin TEXTpsrout
FCL5		FCL5psrin CPS1psrout
SORT		SRT4psrin
SORT5		SRT5psrin
ALGOL5		AGL5psrin
OSLIB		OPLpsrout TLIBpsrin
FCL2		FCL4psrin CPS1psrout
BIT8		BIT8psrin
CID		CID1psrin OPLpsrout
PMD		PMD5psrin CPS1psrout
F45		F451psrin CPS1psrout

Table 4-2. Group 2 Products

Procedure Name	Unique Keywords	Required Files
MSS	SAVELIB	OPLpsrout
CCL		CCL1psrin CPS1psrout OPLpsrout
FSE		OPLpsrout
PLI		PLIpsrin CCG1psrout CPS1psrout
NIP5870		OPLpsrout
NSS		NSS1psrin OPLpsrout
LCS3		LCS3psrin
FCS3		FCS3psrin
MAS	SAVELIB	OPLpsrout
TDU		TDU1psrin
APL2	TERMTYP=termtyp	APL2psrin OPLpsrout
FORM		FORMpsrin
TRACER		OPLpsrout

Table 4-3. Group 3 Products

Procedure Name	Unique Keywords	Required Files
AAM2	SRT4	AAM2psrin
DDL3		DDL3psrin CPS1psrout
FDBF	FTN4	FDBFpsrin CPS1psrout DDL3psrout
CDCS2	DEBUG, SRT4	CDCSpsrin AAM2psrout DDL3psrout OPLpsrout CPS1psrout
COBOL5	TAF, NOCDCS,SRT4	COB5psrin
COBOL5Q	TAF, NOCDCS,SRT4	COB5psrin
DCL		DCL2psrin
IMF1	SRT4	IMF1psrin CPS1psrout DDL3psrout
QU3	SRT4, IMF	QU31psrin DDL3psrout OPLpsrout

Table 4-4. Group 4 Products

Procedure Name	Unique Keywords	Required Files
FTN5		FTN5psrin CPS1psrout CCG1psrout
MSSI	FTN4 BLDMLIB MSAMLIB	MSSIpsrin OPLpsrout
AP1I	FTN4 MEMSIZE	AP1Ipsrin
MMCL	FTN4	MMCLpsrin
AP1L	FTN4 MAPTYPE	MMCLpsrin
CEDIAG		CEDGpsrin OPLpsrout
TOOLS		OPLpsrout

Table 4-5. Group 5 Products

Procedure Name	Unique Keywords	Required Files
RHF		RHF1psrin RHC1psrout OPLpsrout
ITF	NOTRACE	ITF1psrin RHC1psrout OPLpsrout
RHP	SUBSYS=subsys TRACE, DEBUG	RHP1psrin RHC1psrout OPLpsrout
NAM5	NOTRACE	NAM5psrin OPLpsrout
NAM5D		NAM5psrin OPLpsrout
IAF		OPLpsrout
RBF5	NOTRACE	RBF5psrin NAM5psrout OPLpsrout
RBF5D		RBF5psrin NAM5psrout OPLpsrout
RDFEX		OPLpsrout
MCS	TRACE	MCS1psrin NAM5psrout OPLpsrout
TAF	DEBUG, TASKLB	OPLpsrout
CROSS	refer to section 5	CRSSpsrin
CCPPH1	DIAG, REMT, BSTP, XTRAPLS	CCPBpsrin CCPTpsrin CCPRpsrin CCPDpsrin
CCPBLB	XREF	
CCPVAR	VN=vn	
CCPLOAD	GN=gn	

Table 4-6. Group 6 Products

Procedure Name	Unique Keywords	Required Files
NOS2B		OPLpsrout NOSBpsrin

Table 4-7. OIP Options and Associated Build Procedures
(Sheet 1 of 3)

Product Name	Related Build Procedures
ALGOL-60 5	ALGOL5
APL 2	APL2
BASIC 3	BASIC3
COBOL 5	COBOL5, FCL1, FCL2, PMD
Communications Control Program (CCP) 3	CCPPH1, CCPBLB, CCPVAR, CCPLOAD
Conversion Aids Subsystem 3	LCS3, FCS3
CYBER Cross System 1	CROSS
CYBER Databae Control System 2	CDCS2
CYBER Interactive Debug	CID
Data Catalogue 2 Version 2.0	DCL
Data Description Language	DDL3
FORTRAN Database Facility 1	FDBF
FORTRAN Extended 4	FTN4, FCL1, FCL2, PMD
FORTRAN Extended 4 with Interactive Option	FTNTS, FCL1, FCL2, PMD
FORTRAN 5	FTN5, FCL5, PMD, CCG
FTN 4/5 Conversion Aid 1	F45
Full Screen Editor	FSE
High Speed I/O Package	HSIO

Table 4-7. OIP Options and Associated Build Procedures (Sheet 2 of 3)

Product Name	Related Build Procedures
Information Management Facility 1	IMF1
Interactive Facility 1	IAF
Interactive Transfer Facility	ITF, RHC
Link Interface Program under CCP 3	(Part of CCP 3)
Maintenance Package	TOOLS, CEDIAG, FORMAT, SYMPL
Mass Storage Extended	MAS
Mass Storage Subsystem 1	MSS
MCCL V3, MAP Macro Control Library	MMCL
Message Control System 1	MCS
MSSI V3, MAP III	MSSI, AP1I, AP1L
Multi-Mainframe Module 1	MMF
Network Access Method 1	NAM5/NAM5D
NOS Online Manuals	No Build Procedure
NOS 2 Package	LOADER, COMPASS, BAM, BIT8, OSTEXT, UPDATE, FORM, AAM2, TEXT, TEXTIO, BINEDIT, CCL, MDD, RDFEX, OSLIB, NOS, NOS2B, NIP5870, TDU
NOS Scope 2 Station	NSS
PASCAL 170	PASCAL
PL/I 1	PLI, CCG
Printer Support Utility	No Build Procedure
PTF/QTF File Transfer Facilities	RHP, RHC
Query/Update 3	QU3
Remote Batch Facility 1	RBF5/RBF5D
Remote Host Facility Access Method	RHF, RHC

Table 4-7. OIP Options and Associated Build Procedures (Sheet 3 of 3)

Product Name	Related Build Procedures
Sort/Merge 4	SORT
Sort/Merge 5	SORT5
Tracer 1	TRACER
Transaction Facility 1	TAF, COBOL5†
Xedit 3	XEDIT
3270 TIP for CCP	(Part of CCP 3)

†All the necessary supporting object routines are not delivered in the TAF TRANLIB. You must run the COBOL5 installation procedure to add various routines that contain the TAF interface to TRANLIB. Refer to section 5.

STEP 2: MOVING PERMANENT FILES TO THE APPROPRIATE USER NAMES

Some of the files created during the installation must be put in the appropriate permanent file catalog, usually UN=SYSTEMX, LIBRARY, or NETOPS. SYSGEN(FULL) installs a CDC-released version of the permanent files automatically. The following installation procedures produce files that must be relocated:

Procedure	Username	Files
CCPLOAD	NETOPS	Gxxx (change name to NLFFILE)
NAM5/NAM5D	NETOPS	NAMSTRT
	SYSTEMX	NAM,NAMNOGO
IAF	SYSTEMX	IAF,IAFTM,IAFTR
MSS	SYSTEMX	MSS
RDFEX	SYSTEMX	RDF
TAF	SYSTEMX	TAF
	KB100DC	TASKLIB
XEDIT	LIBRARY	XEDITH
APL2	APL0,APL1	Refer to section 5
CDCS	SYSTEMX	CDC
DCL	LIBRARY	Refer to section 5
MCS	SYSTEMX	MCS
NSS	SYSTEMX	SSF
FSE	SYSTEMX	SMF
MAS	SYSTEMX	MAS
	SUBFAM0	MSASCON

The files listed above are placed on the installation user name after the corresponding installation procedure has been executed.

The files to be moved are described under the installation procedure name in section 5. Section 5 also gives general information about subsystem startup.

To move the files, use the SYSGEN(MOVE) command. For information, refer to appendix B.

STEP 3: CREATING A NEW DEADSTART TAPE

To create a new deadstart tape, enter this command:

```
BEGIN(GENDST,INSTALL,SYSTEM=ods,NEW=new,D=d,LIST=list)
```

This command merges the binaries in file PRODUCT with the current system or another deadstart tape to produce a new deadstart file. Refer to appendix B for more information about GENDST.

If you are building on a system with limited disk space, you should execute GENDST followed by RESETP to reduce the size of file PRODUCT.

STEP 4: VERIFYING YOUR INSTALLATION

Verify the installation of products as follows.

- Execute the REPORT procedure to get statistics on all completed installation procedures.
- Execute a verification procedure to determine if the installed product is operational.

The REPORT procedure can be executed after any installation job is completed. Refer to appendix B for information about the REPORT procedure.

The verification procedures cannot be executed until you have completed the installation and have deadstarted the system using your new deadstart tape.

VERIFICATION PROCEDURE

Check the installation of the products with the verification procedures (refer to table 4-8). A verification procedure can be run after the corresponding product is installed in the running system. That is, rebuild the product, create a new deadstart tape, deadstart with the new system, and then run verification jobs. Note that not all products have verification procedures. Also, NAM, CCP, and RHF verifications are more complex than other product verifications. For information on NAM, CCP, and RHF verifications, refer to Verify NAM and CCP and Verify RHF in this section.

To verify that a product is successfully installed, run a job that contains this command:

```
BEGIN,pname,INSTALL,PRID=id.
```

Parameter	Description
pname	pname is the verification procedure name of the product to be verified (refer to table 4-8).
PRID=id	id is 0 to 60. This parameter specifies the printer for verification job output.

The following command calls all verification procedures on file INSTALL (refer to table 4-8) except VCDCS2A, VCDCS2B, VMCS1A, VMCS1B, VRHF, and VMSSI (refer to the descriptions of the CDCS2 and MCS procedures in section 5).

```
BEGIN,VJOBS,INSTALL,PRID=printerid,PACE=n.
```

The parameter printerid can be specified to select any printer you require (the default is 0).

The parameter PACE is useful in delaying job submission in case your system has a limit on the number of jobs in the system. The default for PACE is 15. Four jobs are submitted, then VJOBS does a ROLLOUT,15. before continuing.

Some verification procedures request you to mount tapes. If DISKINS=YES, the proper disk file is automatically used.

Since VJOB submits all verification procedures, procedures that verify unordered products fail.

Table 4-8. Verification Procedures

Verification Procedure Name	Required Files	Comments
VAAM2		
VALGOL5		
VAPL2		
VBAM		
VBASIC3		
VBIT8		
VCDCS2A		
VCDCS2B		
VCID		
VCOBOL5		
VCROSS	CRSSpsrout	
VDDL3		
VFCS3	FCS3psrout	
VFDBF		
VFORM		
VFTN		
VFTNTS		
VFTN5		
VF45		
VIMF1		
VJOBS		Submits all verification jobs except VCDCS2A, VCDCS2B, VMCS1A, VRHF, VMSSI, and VMCS1B.
VLCS3	LCS3psrout	
VMCS1A		MCS must be enabled execute VMCS1A (refer to section 5).
VMCS1B		MCS must be enabled to execute VMCS1B (refer to section 5).
VMSSI		Contains an FTN4=YES/NO parameter. The default is FTN4=NO. Refer to section 5.
VPASCAL		
VRHF		
VPLI		
VQU3		VQU3 must be run as a batch job.
VSORT		
VSORT5		
VSYMPL		

VERIFYING NAM AND CCP

Use the following procedure to verify correct installation of NAM and CCP.

1. Master clear all local and remote network processing units (NPUs). Refer to the NOS 2 Analysis Handbook.

2. Initiate NAM at control point n by entering the following DSD commands at the system console.

```
SUBSYST.
L.ENABLE,NAM,n.
NAMNOGO.
CFO,NAM.GO.
```

   Refer to the NOS 2 Analysis Handbook for more information on the NAMNOGO and the CFO commands.

   If IAF is to be brought up, you must not initiate NAM at control point 1 (n must be 2 or greater).

   When NAM actually starts execution (that is, after NAMI has completed execution), NAM sends its version number to the system dayfile. The message has the following format.

```
NAM VER ver - lvl
```

ver	Version number.
lvl	Programming Systems Report (PSR) summary level.

   When the network supervisor (NS) starts execution, it places in NAM's dayfile its version number and the date and time that both the network configuration file (NCF) and the network load file (NLF) were built. These messages have the following format.

```
NS/     time VER ver - lvl.
NS/     time NCF build date, build time.
NS/     time title.
NS/     time NLF build date, build time.
```

time	Time at which the message was sent to the dayfile.
ver	Product version number and lvl is the PSR summary level.
title	Site-supplied string to the Network Definition Language Processor (NDLP) for the NCF.

   NS proceeds to load all accessible NPUs. NS issues load status messages to the NAM K display (if the NAM K display is assigned to NS) and to the NAM dayfile (refer to the NOS 2 Analysis Handbook).

After the communications supervisor (CS) begins execution, it issues messages to the NAM dayfile having the following format.

```
CS/      time VER ver - lvl.
CS/      time NCF build date, build time.
CS/      time title.
```

time	Time at which the message was sent to the dayfile.
ver	Product version number.
lvl	PSR summary level.
title	Site-supplied string to NDLP for the NCF.

As each NPU is loaded, CS issues messages to the NAM K display (if the NAM K display is assigned to CS) and to the NAM dayfile having the following format.

```
NP/npuname     time NPU: npuname, AC npuid
NP/npuname     time SUPERVISION GAINED
NP/npuname     time CCP VERSION: ver, LEVEL: h, VARIANT: v
```

time	Time at which the message was sent to the dayfile.
npuname	Name of the NPU.
npuid	Node number of the NPU.
ver	CCP version number.
h	Hexadecimal number indicating the level of code in this version of CCP (adjusted by Control Data; not an installation parameter).
v	Hexadecimal number identifying the CCP variant.

When the network validation facility (NVF) starts execution, it sends its version number and the local configuration file (LCF) build date and time to the NAM dayfile. These messages have the following format.

```
NV/      time VER ver - lvl.
NV/      time LCF build date, build time.
NV/      time title.
```

time	Time at which the message was sent to the dayfile.
ver	Product version number.
lvl	PSR summary level.
title	Site-supplied string to NDLP for the LCF.

After NS successfully loads an NPU, CS issues a message to the NAM K display and the NAM dayfile for every logical link defined as active in the network configuration file. The message format follows.

```
LLINK: linkname,EN,RL=3,H-N,npuname1/nid,npuname2/nid.
```

Parameter	Description
linkname	Name assigned to link in the network configuration file.
npuname1	Name of NPU that is connected to the coupler or trunk.
npuname2	Name of NPU at the other end of the logical link.
nid	End node number of logical link (either coupler or NPU node number).

NOTE

The remainder of this procedure determines that the network is running and able to process applications. Terminal Verification Facility (TVF) is used in this procedure. Use of TVF is described in the NAM1/CCP3 Terminal Interfaces Reference Manual.

3. Assign the K display to NAM by typing the following commands at the system console.

   ```
   K,NAM.
   K.ST.
   ```

   The NAM status display should appear on the screen. Refer to the NOS 2 Analysis Handbook for an explanation of the format.

   All applications connected to NAM should appear in the applications section (the column titled APP) of the status display. All front-end NPUs that are logically on in the equipment status table (EST) should appear in the EST section of the display. All enabled/active logical links should appear in the logical link section of the display.

4. Log in from any network-supported terminal, specifying TVF as the application. The user name under which the login is performed must have permission to access TVF. TVF responds with several lines of information about the terminal followed by a prompt for input (..).

5. Enter 2 followed by the message transmission key for the terminal class in use (carriage return for most asynchronous terminals). This initiates the TVF line test. TVF responds as follows.

   ```
   LINE TEST BEGINS
   ..
   ```

6. Enter any character followed by the message transmission key. If your page width is greater than 0, TVF responds by printing a single line composed of the character you entered followed by:

   ```
   TVF TEST COMPLETE
   ..
   ```

7. Enter END to exit TVF. The system responds by issuing a message indicating the time of connection to TVF followed by a prompt for application selection.

   ```
   TVF      CONNECT TIME hh.mm.ss
   termname    -APPLICATION:
   ```

8. Enter BYE to exit the network and log off. The system responds as follows.

   ```
   LOGGED OUT.
   ```

RHF INSTALLATION VERIFICATION

The VRHF procedure on DECKOPL can be used to verify the correct installation of Remote Host Facility (RHF) and permanent and queue file transfer applications. Before executing the procedure, ensure that the logical identifier table (LID) table is set up correctly.

- Enter NC in your EQPDECK for the network access device (NAD).
- Bring up the LID display by entering the LIDOU command.
- Ensure that the physical identifier (PID) (M plus the 2-character machine identifier) matches the PID used on the BEGIN command entered under DIS.
- Ensure that the LIDs LBK and LB2 are loopback LIDs under the host PID.

Under DIS, execute VRHF with the following command, using the correct physical ID and NAD parameters for the mainframe where the test is to execute.

```
BEGIN,VRHF,INSTALL,PID=Mid,NADADDR=adr,CHANNEL=ch.
```

Parameter	Significance
id	Machine identifier.
adr	NAD address.
ch	NAD channel.

Success of the verification is indicated by the message VERIFY GOOD in the output listing of the RHFVJOB job.

SPECIAL PRODUCT INSTALLATION INFORMATION 5

This section gives special information for the customized installation of NOS and its products. The products are listed in alphabetical order according to their installation procedure name. If special information about a product does not exist, the product installation procedure is not listed. Here is a list of the installation procedures discussed in this section:

AAM2
APL2
BAM
BASIC3
CCL
CCP/CROSS (CROSS, CCPPH1, CCPBLB, CCPVAR, CCPEDIT, CCPLOAD, CCPPURG)
CDCS2
CID
COBOL5 and COBOL5Q
DCL
FDBF
FSE
FTN4, FTNTS, and FTN5
IAF
IMF1
ITF
LCS3 and FCS3
LOADER
MAS
MCS
MSS
MSSI, MMCL, AP1I, AP1L
NAM5
NOS
NOS2B
NSS
QU3
RBF5, RBF5D
RHF
RHP
SORT
TAF
TDU
TEXT and TEXTIO
UPDATE
XEDIT

To get listings of the installation procedures, refer to the DECKLIS procedure in appendix B.

The installation of controlware differs from product installation procedures; controlware installation is described in appendix D.

SUBSYSTEM INITIATION

SUBSYSTEM STARTUP FILES

When you install a subsystem, you must ensure that there is a startup procedure file for initiating the subsystem. (Table 5-1 lists all the subsystems.) Startup files must be stored as indirect access files on user name SYSTEMX. They must also be named using the following format:

subxxxx

sub	The 3-character name for the subsystem (for example, CDC).
xxxx	An optional 0- to 4-character extension to the name (for example, CDCS2).

DECKOPL installation procedures create startup procedures for all subsystems and place them as indirect access files on the installation user name INSTALL. The files are named with the appropriate 3-character subsystem name. SYSGEN(FULL) places the released versions of all subsystem startup procedures on the system user name SYSTEMX. If you modify the procedures, you can use the SYSGEN(MOVE) procedures to move the files. Refer to appendix B for more information about SYSGEN(MOVE).

SUBSYSTEM ENABLE AND DISABLE COMMANDS

In addition to the procedure file creation and placement, you must issue ENABLE commands to tell the system which control point the subsystem is to operate as well as what system functions are required. You can enter the ENABLE (or DISABLE) commands at the system console or you can add them to the IPRDECK. Table 5-1 lists the necessary ENABLE and DISABLE commands.

AUTOMATIC SUBSYSTEM INITIATION

If you want a subsystem to be automatically initiated when you enter the DSD AUTO or MAINTENANCE commands, you should do the following:

- Ensure that the startup procedure file has the 3-character name of the subsystem and is stored as an indirect access file on user name SYSTEMX.

- Enable the subsystem and any other system functions by adding the appropriate ENABLE entries to the IPRDECK.

Table 5-1 summarizes the subsystem procedure names and the appropriate ENABLE commands necessary for initiating the subsystem.

Table 5-1. Subsystem Initiation
(Sheet 1 of 2)

Installation Procedure	Subsystem	Procedure Name	ENABLE Commands
CDCS2	CDC	CDC	ENABLE,SCP. ENABLE,CDC,n.
FSE	SMF	SMF	ENABLE,SCP. ENABLE,SMF,n.
IAF	IAF	IAF,IAFTM,IAFTR	ENABLE,SCP. ENABLE,IAF.
MCS	MCS	MCS	ENABLE,SCP. ENABLE,MCS,n.
MAS	MAS	MAS	ENABLE,SCP. ENABLE,MAS,n. ENABLE,MASTER MAS. ENABLE,FILE STAGING.
MSS	MSS	MSS	ENABLE,SCP. ENABLE,MASTER MSS. ENABLE,FILE STAGING. ENABLE,MSS,n.
MSSI	MAP	MAPCMI, MAPECS, MAPCH	ENABLE,SCP. ENABLE,USER EXTENDED MEMORY. ENABLE,MAP,n.
NAM5/NAM5D	NAM	NAM,NAMNOGO	ENABLE,SCP. ENABLE,NAM,n.
NSS	SSF	SFF	ENABLE,SCP. ENABLE,SSF,n.
NOS	MAG	MAG	ENABLE,MAG.
RBF5/RBF5D	RBF	No procedure, initiated by NAMI	RBF initiation is controlled by NAMI

Table 5-1. Subsystem Initiation
(Sheet 2 of 2)

Installation Procedure	Subsystem	Procedure Name	ENABLE Commands
RDF	RDF	RDF	ENABLE,RDF.
RHF	RHF	No procedure	ENABLE,SCP. ENABLE,RHF,n.
TAF	TAF	TAFPRC	ENABLE,SCP. ENABLE,SUBCP. ENABLE,TAF,n.

NOTES:

- RDF and MAG are always added to the deadstart file for system maintenance purposes. All other procedures are placed only on permanent files.
- MAS, MSS, and RDF may require additional IPRDECK entries.
- For more information about the IPRDECK, ENABLE and DISABLE commands, and subsystem startup, refer to the NOS 2 Analysis Handbook.
- Enable either IAF or RDF, not both.
- If you do not want to store the startup procedure files on user name SYSTEMX, you can put the procedure files on the deadstart tape and add *PROC directives to your LIBDECK. For more information, refer to the NOS 2 Analysis Handbook.

AAM2 - CYBER RECORD MANAGER ADVANCED ACCESS METHODS VERSION 2

This subsection describes these installation options for AAM2:

- Unique Parameters
- USER File Directives
- Additional Procedures

UNIQUE PARAMETERS

SRT4 To assemble code to interface with Sort/Merge 4 rather than Sort/Merge 5, specify the keyword SRT4 on the call to AAM2.

USER FILE DIRECTIVES

If you want to gather additional file statistics, include the Update directive *DEFINE STATS in file USER. (Without this directive, the system gathers only normal file statistics.)

ADDITIONAL PROCEDURES

AAM2 includes one system compression/decompression routine. You can add up to 53 additional compression/decompression routines as system routines. Encapsulate each added routine and modify the capsule OPNM$AA.

Each routine must have an entry point of the form CMPR$nn (nn is two decimal digits within the range 11 through 63). The first added routine's entry point name must be CMPR$11, the second routine's entry point name must be CMPR$12, and so forth. The entry point must be the second word (word 1) of the routine.

The first three words of each routine must have the format shown in table 5-2.

Table 5-2. Format of First Three Words of Compression/Decompression Routines

Word	Bits	Contents
0	59 through 18	Entry point name, 6-bit display code, left-justified, zero fill.
	17 through 0	1.
1	59 through 18	0.
1	17 through 0	Starting address of compression code.
2	59 through 18	0.
	17 through 0	Starting address of decompression code.

An example of the construction of a single site-added compression/decompression routine follows:

```
         IDENT
         ENTRY   CMPR$11
         VFD     42/0LCMPR$11,18/1
CMPR$11  VFD     42/0,18/COMPRES
         VFD     42/0,18/EXPAND
         .
         .
         .
COMPRES  BSSZ      1
         .
         .
         .
         EQ      COMPRES
         .
         .
         .
EXPAND   BSSZ      1
         .
         .
         .
         EQ      EXPAND
         END
```

The CYBER Loader requires standard relocation for fast dynamic loading of capsules; therefore, construct the VFD statements as shown in the preceding example. A return jump to the address specified in word 1 or 2 of the routine affects the execution of the compression or decompression code.

For each added routine, add an entry to the capsule name table in deck OPNMDAA. The macro GENTBL (also part of OPNMDAA) generates the table entry and has the following format.

```
GENTBL name
```

Parameter	Description
name	Entry point name specified in word 0 of added routine.

Specify table entries in consecutive, ascending numerical order. For example, to add three routines, make the following change to OPNMDAA.

```
*B OPNMDAA.329
          GENTBL CMPR$11
          GENTBL CMPR$12
          GENTBL CMPR$13
*C OPNMDAA,DICODAA,CWEOR1,OPENDAA
```

To add one additional compression/decompression routine, execute a job similar to the following. Either it must be a system origin job, or you must have system origin privileges and have DEBUG on.

```
        Job                                              Comments

    job command.
    USER,INSTALL,INSTALL.
    BEGIN,PRDIN,INSTALL,PRDNAME=AAM2,DISK=0.
    UPDATE,K.
    RFL,65000.
    COMPASS,A,I,S=TXTCRM,S=IPTEXT.         Assembles OPNMDAA and DICODAA.
    SYMPL,ET=T,I,S=TXTCRM,S=IPTEXT.        Compiles OPNMDAA.
    COMPASS,A.                             Assembles routine being added.
    RETURN,COMPILE.
    GROUP,$AAM$$CTL$.                      Encapsulates the modified capsule OPNM$$AA
    CAPSULE,$OPNM$$AA$.                    (deck OPNMDAA) and the new compression
    CAPSULE,$CMPR$$11$.                    capsule.
    LDSET,OMIT=$SETUP.$/$RM$$SYS=$.
    LOAD,LGO.
    NOGO,NEWCAP.
    COMMON,SYSTEM.                         User must be validated to access common files.
    GTR,SYSTEM,OLD.ULIB/AAMLIB
    LIBEDIT,B=0.
    LIBGEN,F=NEW,P=AAMLIB.
    SYSEDIT.
    --eor--
    *IDENT
        Update directives to modify OPNMDAA.
    *C OPNMDAA,DICODAA,CWEOR1,OPENDAA
    --eor--
        Compression/decompression routine
        being added (COMPASS).
    --eor--
    *DELETE CAP/OPNM$AA
    *FILE NEWCAP
    *BEFORE *,CAP/*
    --eor--
    *FILE AAMLIB
    --eoi--
```

APL2 - APL VERSION 2

The installation of APL2 consists of building APL2 and then installing APL2 permanent files. This subsection describes these installation options for APL2:

- Unique Parameters
- APL Entry Message
- SYSGEN Functions
- APL Validation Requirements

UNIQUE PARAMETERS

TERMTYP=ttype You can specify TERMTYP=ttype on the call to the APL2 installation procedure. ttype defines the default terminal type. Refer to your listing of DECKOPL for allowable parameter values. The default for ttype is APLAS.

APL ENTRY MESSAGE

If you want to provide an APL entry message (a one-line message displayed when a user activates APL with a command), create a file named MESSAGE and make it local just prior to the BEGIN,APL2,INSTALL... command. The file must contain only a one-line message which cannot exceed 80 characters.

SYSGEN FUNCTIONS

Only the APL loader can be captured on a deadstart tape. Except for the loader, APL2 runs from a set of permanent files.

SYSGEN(FULL) installs all APL files on user names APL0 and APL1. If you customize the APL files, install the modified files by performing these steps:

1. Before running the APL2 build procedure, execute these commands from your installation user name:

```
PURGE(PFGAPL2)
DEFINE(PFGAPL2/CT=S)
RETURN(PFGAPL2)
```

2. Run the APL2 procedure. Then enter these commands at the system console:

```
X.DIS.
USER(INSTALL,INSTALL)
ATTACH(PFGAPL2)
SYSGEN(APL2)
```

Once the SYSGEN(APL2) command is completed, the user names APL0 and APL1 will have the updated APL permanent files.

APL VALIDATION REQUIREMENTS

User names APL0 and APL1 are required for running APL. If you have no validation file, SYSGEN will automatically create the user names and assign default passwords. If you have a validation file, you must create user names APL0 and APL1 with the limits shown in table 5-3.

If you want to change the default passwords for the two user names (APL0 and APL1), use the MODVAL command. Refer to the Administration Handbook for information about using MODVAL and about changing validation files. If you want to use SYSGEN to reload files on these user names, you must temporarily set the passwords to the default values or you must update SYSGEN.

Table 5-3. Recommended Limits for APL0 and APL1

Resource or Capability Mnemonic	User APL0		User APL1	
	Keyboard Entry	Converted Value	Keyboard Entry	Converted Value
MT	3	3	3	3
RP	2	2	2	2
TL	77B	Unlimited	77B	Unlimited
CM	40B	2037B	40B	2037B
DB	5B	10	5B	10
FC	7B	Unlimited	7B	Unlimited
CS	4B	4096	4B	4096
FS	6B	192	6B	192
PA	EVEN	Even	EVEN	Even
RO	37B	System	37B	System
PX	HALF	Half	HALF	Half
TT	TTY	TTY	TTY	TTY
TC	NORMAL	Normal	NORMAL	Normal
IS	NULL	Null	NULL	Null
MS	6B	25088	6B	25088
DF	73B	1008	73B	1008
CC	77B	Unlimited	77B	Unlimited
CP	77B	Unlimited	77B	Unlimited
LP	77B	Unlimited	77B	Unlimited
PT	77B	Unlimited	77B	Unlimited
EC	0B	0B	0B	0B
SL	77B	Unlimited	77B	Unlimited
CN				
PN				
DS	3B	1536	2B	1024
SC				
DT				
SP				
UP				
PW	APL0	APL0	APL1	APL1
AW	†	††	†	††

†Entry is CASF, CAND, CSRP, CPWC, CLPF, CSPF, CCNR.
††Value is 00000000000000000755.

BAM - CYBER RECORD MANAGER BASIC ACCESS METHODS VERSION 1

The following installation parameters for BAM are defined in the Update common decks /CMNTXT/ and /TXTCRM/. Assemble deck TXTCRM to obtain a listing of the common decks.

Parameter	Default	Significance
#LBLIM#	10D	Number of words in tape label buffer. Because each user label requires 9 words, set LBLIM to 9m+1; m is the maximum number of file header (HDRn) labels allowed. Minimum value is 10 words.
#CMU# #NOCMU#	1 1	Specifies use of compare and move unit (CMU) instructions in routine MOVE$RM. To remove the CMU code, delete the definition of CMU. To remove the no CMU code, delete the definition of NOCMU. If CMU and NOCMU are both defined, CYBER Record Manager determines at run time which MOVE routine to use by checking the CMU flag in RA.CMU. The use of CMU instructions reduces the execution time of a program using CYBER Record Manager for records of over 40 characters. The use of CMU instructions in programs to be executed on models 835, 840, 845, 850, 855, or 860 is not recommended.

BASIC3 - BASIC VERSION 3

The following installation parameter for BASIC3 is defined in deck BASCOMP. Assemble this deck to obtain the Update sequence number required to change the released value.

Parameter	Default	Significance
BDFLT	1.0	Array base; can be any nonnegative value expressed as a real value.

The following parameters are defined in common deck LIPARAM. Assemble deck BASCARD to obtain a listing of LIPARAM.

Parameter	Default	Significance
MESSAG	1	Flag indicating whether BASIC issues time and memory use dayfile messages. A value of 0 inhibits issuing of messages; a value of 1 enables issuing of messages.
IP.AS	0	Flag indicating default character set mode. A value of 0 indicates normal (non-ASCII) mode (user must specify AS on the BASIC statement to override the default); a value of 1 indicates ASCII mode (user must specify AS=0 on the BASIC statement to override the default).
IP.BL	0	Flag indicating burstable listing. A value of 0 indicates nonburstable listing (user must specify BL on BASIC statement to override the default); a value of 1 indicates burstable listing (user must specify BL=0 on the BASIC statement to override the default).

CCL - CYBER CONTROL LANGUAGE VERSION 1

This subsection describes these installation options for CCL:

- Installation Parameters
- Additional Procedures

INSTALLATION PARAMETERS

The following installation parameters are located on deck CCL. Because installation procedures use the default values, changing the values is not recommended.

Parameter	Default	Significance
IP.FPC	10	Maximum number of characters in a keyword for a procedure call or procedure header directive. Maximum value is 10.
IP.SCS	40	Maximum number of characters for default and actual values. Maximum value is 80.
IP.LCS	10	Maximum number of characters in a label string. Maximum value is 10.
IP.PNL	50	Procedure nesting limit. Maximum value is 1023.
IP.FP	50	Maximum number of keywords in a procedure call or procedure header directive. Maximum value is 500.
IP.DPF	1	Flag indicating logical existence of default procedure file name. Value / Definition 0 No default procedure file name. 1 Procedure file name defaults to value of IP.DPFN.
IP.DPFN	PROCFIL	Default procedure file name.
IP.TAPO	1	Flag indicating whether a procedure can reside on tape. Value / Definition 0 Procedure file cannot reside on tape. BEGIN hangs in RECALL if execution from tape is attempted. A value of 0 decreases the execution size of CCL by 700_8 words for BEGIN, REVERT, WHILE, and ENDW. 1 Procedure file can reside on tape.

Parameter	Default	Significance
IP.EXP	100	Number of operands and operators allowed in a CCL expression. For each unit that this parameter is decreased from 100, the execution size of CCL is reduced by 2 words.
IP.ATT	1	Flag indicating whether the system searches the user name's permanent file catalog, if the requested procedure file is not local. Value — Definition 0 — Permanent file catalog not searched. 1 — Permanent file catalog searched. In order to attach the requested procedure file, the system searches the indirect access files first and then the direct access files.
IP.NPV	6	Value used in the calculation of the size of the pattern value table (PVT). The PVT stores the checklist entries for each parameter in the procedure headers. The following formula determines the size of the PVT in words. PVT = IP.NPV x IP.FP x 2
IP.RLD	1	Flag indicating whether the system does a sequential or random search of a library to find the requested procedure. A random search is usually faster than a sequential search. Value — Definition 0 — Search library sequentially. 1 — Search library randomly by using the library directory.
IP.SCL	150	Maximum length in characters of lines in a procedure. Any restrictions as to the length of a command remain in effect, but a comment following the command terminator may extend to the length specified by IP.SCL.

ADDITIONAL PROCEDURES

CCL consists of three absolute overlays with entry point names and verb table entries for each CCL verb (command). Here are the CCL verbs and overlays:

Overlay	Verbs
CCLBRWE	BEGIN, REVERT, WHILE, ENDW
CCLIFES	IF, IFE, ELSE, ENDIF, SKIP
CCLDS	DISPLAY, SET

If a CCL verb must be changed because of a conflict with an existing program on the deadstart file, change both the entry point name and the verb table entry in the CCL overlay CCLBRWE.

CCP - CYBER CROSS SYSTEM AND COMMUNICATIONS CONTROL PROGRAM

This subsection describes the following:

- Hardware Requirements
- Build Steps Description
- Binary Installation Option
- General Build Step Call
- Cross-Cross System Installation
- CCPPH1 Build Step
- CCPBLB Build Step
- CCPVAR Build Step
- CCPEDIT Build Step
- CCPLOAD Build Step
- CCPPURG Build Step
- CCP/CROSS Permanent Files
- CCP System Definition
- CCP Variant Load Module Definition
- CCP/CROSS Installation Examples

HARDWARE REQUIREMENTS

A field length of 110000_8 is required to build CCP. The following equipment configuration is the minimum required to execute CCP:

- One 2550-2, 2551-1, 2551-3, or 2551-4 Host Communication Processor, consisting of at least the following items:
  - One multiplexer loop interface adapter.
  - One loop multiplexer.
  - One cyclic encoder board.
  - One CYBER communications coupler.
  - One memory unit.
  - One 8K micromemory board.
- One communications line adapter (CLA), either a 2560-1 synchronous CLA or a 2561 asynchronous CLA.
- Total 2550 memory of at least 65K.

Assign the CLA slots in the loop multiplexer in order of decreasing line transmission speeds. For example:

Speed	Slot Assignment
9600-bit per second (bps) line	Slot 1 (leftmost slot)
9600-bps line	Slot 2
2400-bps line	Slot 3
300-bps line	Slot 4
150-bps line	Slot 5

BUILD STEPS DESCRIPTION

The CCP/Cross installation procedures consist of seven sequential build steps. The following build step descriptions are listed in their proper execution sequence.

Build Step	Description
CROSS	Updates the Cross program library on the CRSS source file with corrective code from file CODEOPL and with user corrective code from file UCRS; compiles the updated binaries for use by the CCP build steps; writes an updated version of CRSS file. If you do not want to install CCP, skip the remaining build steps.
CCPPH1	Updates the program libraries on the CCPB, CCPD, CCPT, and CCPR files with corrective code from file CODEPL and then merges the updated program libraries into file PCMB; creates updated program libraries of CCP (PCCP), online diagnostics (PDGN), the 3270 TIP† (PBST), and remote concentrator products (PREM) and any other user-supplied program libraries; updates PCMB with temporary user-supplied corrective code from file UCCP and generates the phase 1 (micromemory) and dump load modules on file ZMUX; creates EXPAND and Autolink binaries; builds system autostart (SAM) load module; writes updated versions of CCPD, CCPT, and/or CCPR.
CCPBLB	Updates the PCMB program library with temporary user-supplied corrective code from file UCCP and generates the CCP object code library (BCMB); writes an updated version of CCPB. This build step is also called the CCP full compile and assembly build step.
CCPVAR	Updates the PCMB program library with temporary user-supplied corrective code from file UCCP and generates a CCP variant load module (Zvvv) and program initiation control block (PICB) file (Ivvv) from the BCMB file according to the user-specified variant definitions in file USERBPS; writes Vvvv. This build step should be repeated for each NPU in the network.
CCPEDIT	Patches a CCP variant load module. This build step is not part of the normal build process but allows the use of the MPEDIT utility of Cross.
CCPLOAD	Updates the PCMB program library with temporary user-supplied corrective code from file UCCP and generates a NAM network load file (Gzzz) via program LFG (refer to the NOS 2 Analysis Handbook). The load file includes the phase 1 and dump load modules (file ZMUX) and system autostart load module (file ZSAM) from step CCPPH1, and the variant load modules ($Zvvv_i$) and PICBs (Ivvv) from step CCPVAR.
CCPPURG	Purges the noncritical permanent files created by the other build steps. It does not purge the load file from build step CCPLOAD and the user-supplied files. This build step is not required; it is only a cleanup utility. However, since previous build steps do not purge the noncritical permanent files, it is suggested that CCPPURG be run to make more disk space available.

The final result of the CCP/Cross build steps is the generation of a NAM network load file. Figure 5-1 illustrates the build step dependencies. Figure 5-2 illustrates the relationship of the load file to the release files and the other files critical to the CCP build process. Figure 5-3 illustrates the relationship of build steps to critical files and tapes involved in CCP installation.

†Terminal Interface Program.

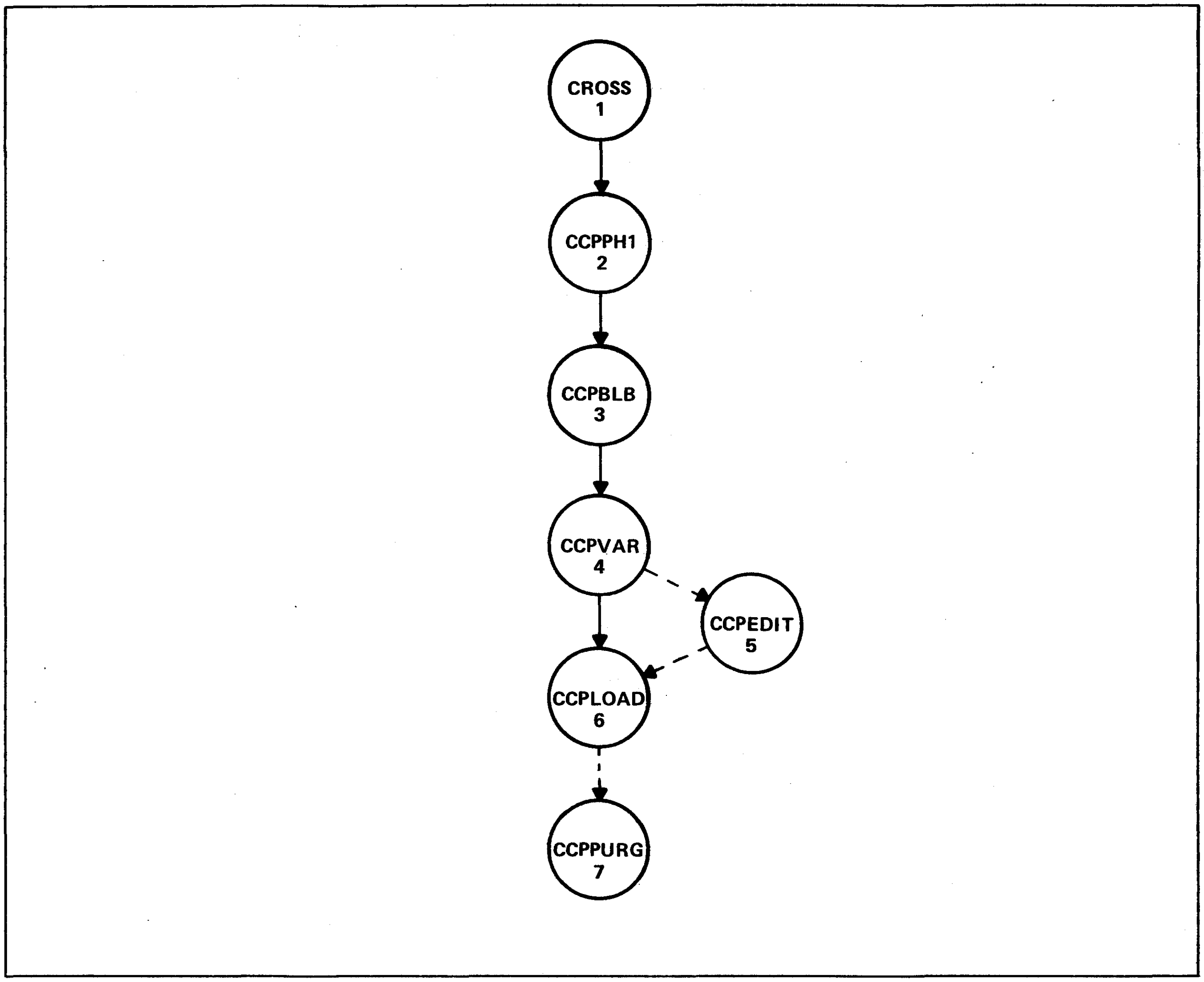

Figure 5-1. CCP/CROSS Build Step Dependencies

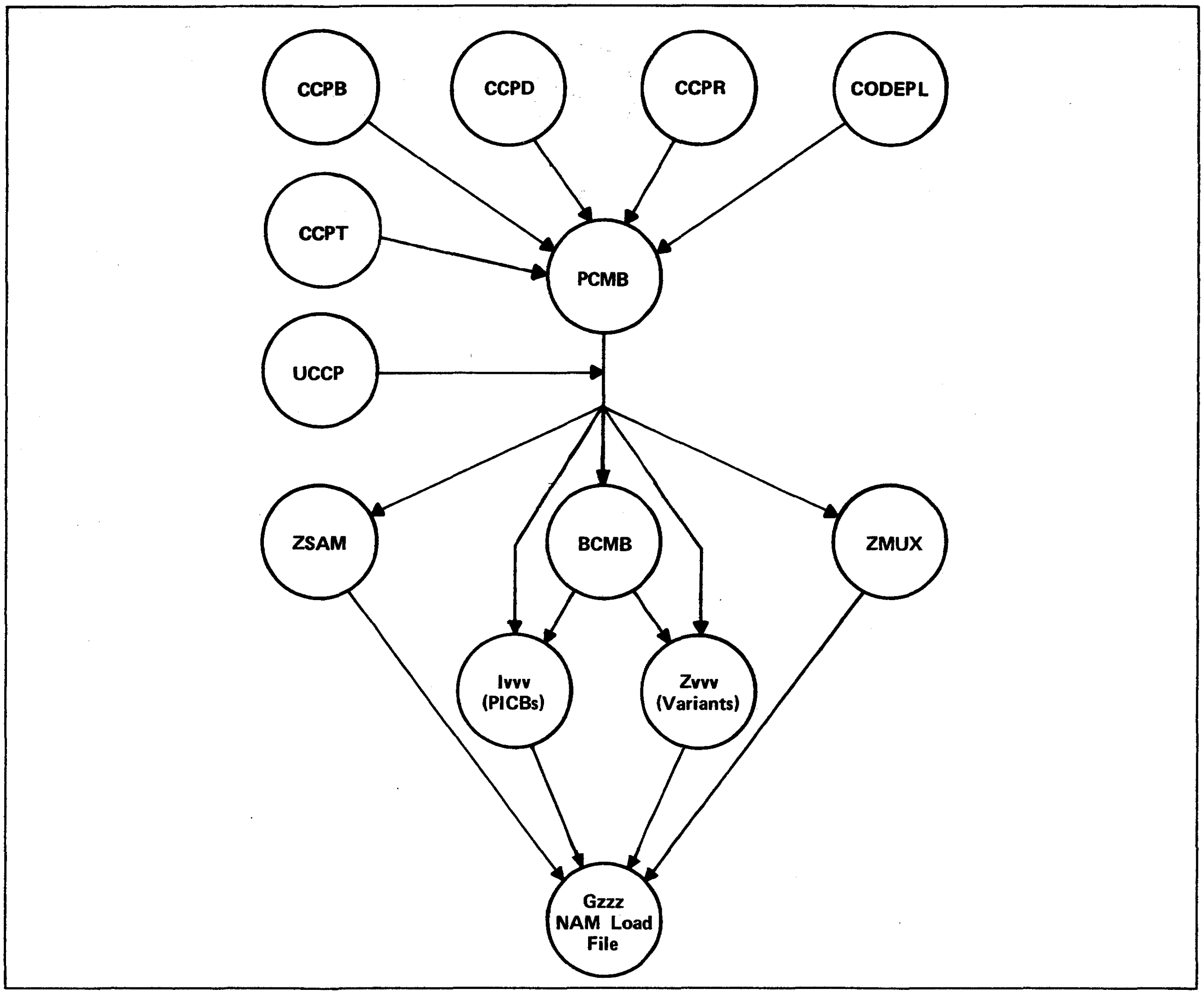

Figure 5-2. CCP File Dependencies

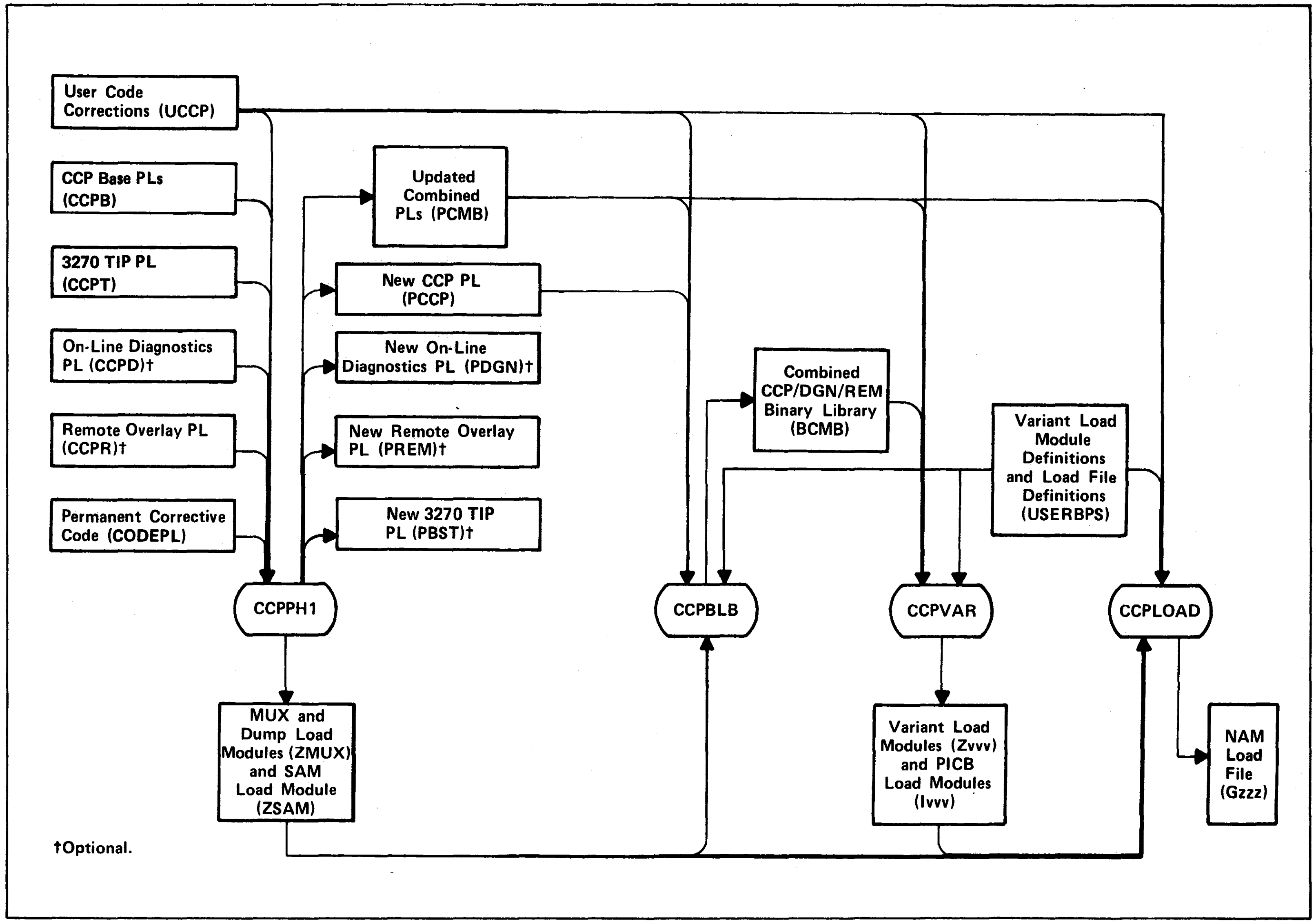

Figure 5-3. Integration of Program Libraries in CCP Build Process

BINARY INSTALLATION OPTION

To speed up the process of installing CCP/Cross, Control Data provides the following prebuilt files on the permanent file tapes:

- All permanent files necessary to begin the build process with the CCPVAR step.
- Several CCP variants and a load file. The Software Release Bulletin (SRB) describes these variants.

Thus, you can skip the first three steps of the build process (the CROSS, CCPPH1, and CCPBLB steps) if these two conditions are true: you do not want to add user code to the the omitted build steps and you do not need to add online diagnostics or the 3270 TIP. (If you ordered the Remote Concentrator Products, they are included in the CCP files.)

To perform a CCP binary installation, enter:

```
SYSGEN(SOURCE,CCP)
```

This command loads all CROSS and CCP permanent files and the USERBPS file to your user name. If you can use both the CROSS and CCP files, modify the LFD and VRD statements in file USERBPS and begin your CCP/Cross installation with the CCPVAR step. If you can use only the CROSS binaries, you should create your USERBPS file and begin your installation with the CCPPH1 step.

NOTE

Complete CCP/Cross and variant build listings (as provided by the LIST=PF option) are provided on the permanent file tapes in the CCP/Cross and variant release files.

GENERAL BUILD STEP CALL

All CCP/Cross build steps are called by the BEGIN command. Descriptions of each of the seven sequential build step procedures, including the required BEGIN parameters, are in subsequent subsections. Table 5-4 summarizes the tape and disk file requirements of the build steps.

Table 5-4. CCP/Cross Tape and Disk File Requirements

Build Step Order	Build Step Name	Input Files Generated by Previous Step	User Input Files	Permanent Files Created	Optional Permanent Files Created
1	CROSS	UCRS	Addd USERCHG CODEPL	LCRB PCRS	
2	CCPPH1		UCCP USERCHG CODEPL	ZMUX PCMB PCCP ZSAM Addd SMUX	LIMC LMFB PDGN PREM PBST LSAM
3	CCPBLB	PCMB ZMUX ZSAM SMUX Addd	UCCP USERCHG USERBPS	BCMB	LFCA
4	CCPVAR	PCMB BCMB SMUX	UCCP USERBPS USERCHG	Zvvv Svvv Ivvv	Lvvv
5	CCPEDIT (optional)	Zvvv Svvv	UEDZ USERCHG	Zyyy Syyy	
6	CCPLOAD	PCMB ZMUX ZSAM Zvvv Ivvv	UCCP USERBPS USERCHG	Gzzz	
7	CCPPURG (optional)		USERCHG		

When you are building CCP/Cross from an auxiliary disk pack, use the PACKNAM command (refer to the NOS 2 Reference Set, Volume 3) before calling the CCP/Cross procedure.

The format of the BEGIN command follows:

BEGIN,pname,INSTALL,p_1,p_2,...,p_n.

pname is the name of the build step procedure. The common parameters LIST, NOECS, NOWRT, and NOPURGE can be set in file COMMOD with the values described below. Refer to appendix B for information about modifying parameters in file COMMOD.

Parameter	Description
p_1=p_n	Build step parameter. The order-independent format should be used.
BSTP=bst	Specifies whether the 3270 TIP program library is present; used only with CCPPH1. If this parameter is omitted, BSTP=NO is assumed.
DIAG=diag	Specifies whether online diagnostics are present (CCPD); used only with CCPPH1. If this parameter is omitted, DIAG=NO is assumed.
GN=file	Specifies load file name. The user supplies the 3-character, alphanumeric file name; used only with CCPLOAD.
LIST=option	Specifies whether the build step creates a listing, saves the listing as a permanent file on disk, and/or assigns the listing to OUTPUT. Do not specify this parameter for the CCPLOAD build step. The default is LIST=NO. YES — Listing is assigned to OUTPUT. For the CCPVAR build step, listing is assigned to OUTPUT and is copied to the new release file. PF — Listing is stored as a permanent file on disk; later it is copied to the new release file and purged from the disk. BOTH — Listing is stored as a permanent file as well as assigned to OUTPUT; later it is copied to the new release file and the permanent file is purged. NO — No listing is created.
NEW=yyy	Specifies new CCP variant name for patched load module; used only with CCPEDIT. Supply the 3-character alphanumeric name.
NOECS	Specifies for build steps CCPPH1 and CCPBLB that extended memory is not used. Supply this parameter only when extended memory is down or when you do not want to use extended memory even if it is available. This parameter is not required if there is no extended memory or if fewer than 77000_8 words of extended memory are available.
NOPURGE	Specifies that routine DRTBAT1 does not purge files PCCP, LIMC, LMFB, LSAM, and LFCA. If NOPURGE is omitted (the default value), the files are purged.
NOWRT=nn	Specifies whether routine DRTBAT1 writes new files. The default is NOWRT=NO. If you want the DRTBAT1 routine to write new files, specify NOWRT=YES.

Parameter	Description
OLD=xxx	Specifies CCP variant to be patched; used only with CCPEDIT. Supply the 3-character alphanumeric name.
PC=pc	Controls the inclusion of code to update the released program libraries from the previous release to the PSR summary level of the current release. If the product program library was included with the current installation materials, PC=NO should be used. The default is PC=NO.
REMT=remt	Specifies whether remote concentrator products are present (CCPR); used only with CCPPH1 (Link Interface Program). If this parameter is omitted, REMT=NO is assumed.
VN=vvv	Specifies a variant name that matches the variant name in the VRD definition in USERBPS; used only with CCPVAR.
Vx=vvv	Specifies the variant name that was used in CCPVAR; used only with CCPPURG. x is an integer within the range 1 through 10.
XREF=xref	Specifies whether the build step generates a cross-reference listing of the Pascal source of CCP; used only with CCPBLB. If this parameter is omitted, XREF=NO is assumed.
XTRAPLS=xtrapls	Specifies whether extra program libraries are to be merged into the PCMB; used only with CCPPH1. If this parameter is omitted, XTRAPLS=NO is assumed.

SECURITY CHARACTER PARAMETERS

The specification of a security character for a particular TIP activates the secure login feature. This feature guarantees that the terminal user can request a connection to the Network Validation Facility (NVF) regardless of any action by a host program. As a result, the login information the user enters remains secure. The installer is responsible for maintaining the integrity of the network configuration files (NCF and LCF) such that the NVF login or autologin is not subverted.

When the terminal user enters the security character in a specific sequence (refer to the NOS 2 Reference Set, Volume 3), CCP terminates any current connection and either reconnects the user to the host computer or prompts the user to select or connect to a host computer.

The security character must be a 7-bit character (specified as a hexadecimal number) that is within the code set of the terminal specified in ASCII. The character is restricted to the values \$03-\$1F, \$21-\$2F, \$3A-\$3C, \$3E-\$40, \$5B-\$60, and \$7B-\$7E. The security character must not be the same value as specified for the abort block, backspace, user break 1, user break 2, cancel, control, end-of-line, or end-of-block character. Refer to the Network Definition Language Reference Manual for the default values for these characters. For any sub-TIP for which a security character is not specified (that is, value equals \$00), the secure login is not activated.

Here are the parameter values in deck SECURITY the CCPB PL.

Parameter	Default Security Character	Terminal
SCAN2741	$00	Asynchronous non-2741 terminals
SCA2741	$00	Asynchronous 2741 terminals
SCB3270	$00	IBM 3270 terminals
SCMD4A	$00	Mode 4A terminals
SCMD4C	$00	Mode 4C terminals
SCHPOST	$00	HASP postprint terminals
SCHPRE	$00	HASP preprint terminals
SCB2780	$00	IBM 2780 terminals
SCB3780	$00	IBM 3780 terminals
SCXPAD	$00	X.25 package assembly/disassembly (PAD) terminals
SCXUSER	$00	X.25 user-defined terminals

CROSS-CROSS SYSTEM INSTALLATION

The following build step generates updated program binaries for all Cross programs and installs those programs needed for the following CCP build steps. Refer to General Build Step Call, earlier in this section, for descriptions of the parameters.

```
BEGIN,CROSS,INSTALL,LIST=option,PC=pc.
```

The CROSS build step uses the following files for input.

File	Description
USERCHG	User/charge file. It is optional for all build steps (refer to CCP/Cross Permanent Files, later in this section).
CRSSpsrin	Cross release file.
UCRS	Optional site corrective code (refer to CCP/Cross Permanent Files, later in this section). For a description of the Cross installation parameters that can be changed, refer to Installation Parameters for CROSS, later in this subsection.
CODEPL	Cross corrective code, if any, that affects the resulting Cross binaries but is not placed in the program library on the output file.

The CROSS build step creates the following output files.

File	Description
CRSSpsrout	New CRSS file.
LCRB	Cross system listings (if requested).
APAS	MP17 Pascal compiler.
AASM	Cross macro assembler.
AMAC	Macro assembler text file.
AFMT	Pascal binary output formatter program.
ASMAS	Cross micro assembler.
AXRF	Pascal cross-reference program.
ALNK	MPLINK program.
AEDT	MPEDIT program.
ALIB	MPLIB program.
ACYP	CYBER 170, CYBER 70, and 6000 Computer Systems Pascal compiler.

Examples:

You can execute the CROSS build step by using any of the following methods.

BEGIN,CROSS,INSTALL.

The build step generates no listings and installs the corrective code.

BEGIN,CROSS,INSTALL,LIST=YES.

The CROSS build step copies its listings (LCRB) to OUTPUT. By default, the build step installs the corrective code.

BEGIN,CROSS,INSTALL,LIST=PF,PC=NO.

The corrective code on file CODEPL (Update-formatted) makes corrections to the file CRSSpsrout.

The following parameters are in deck CROSS.

Identifier	Parameter	Default	Significance
XSYA127.6	MAXGLBL	1535	Maximum number of global symbols minus one.
XSYA127.7	HGHPAGE	55	(SYMTBSIZ/32)-1.
XSYA127.8	SYMTBSIZ	1792	Size of in-core symbol table.
XSYA127.9	VARPAGE	47	MAXGLBL/32.
XSYA127.406	SYMTBSIZ	1792	Size of in-core symbol table.
XSYA127.407	MAXGLBL	1535	Maximum number of global symbols minus one.

The number of entries in the in-core symbol table in the release version of the Pascal compiler is 1792. This version of the compiler has a corresponding maximum number of global symbol definitions of 1536 and an execution field length of 77000_8 central memory (CM) words. Some programs require a Pascal compiler that accommodates more than 1536 global symbol definitions; for example, CCP requires 6144 global symbols. Increasing the size of the global symbol table without increasing the in-core symbol table, however, results in a significant increase in compilation time. Further, an increase in the number of CM words must accompany any increase in the size of the in-core symbol table (4 CM words per symbol table entry).

CCPPH1 - CCP PHASE 1

The following build step generates a combined base program library for CCP, 3270 TIP, online diagnostics, and remote concentrator program libraries. It also creates the multiplexer firmware and the dump load module. Refer to General Build Step Call, earlier in this section, for descriptions of the parameters.

```
BEGIN,CCPPH1,INSTALL,DIAG=diag,REMT=remt,BSTP=bst,NOECS.
```

The CCPPH1 build step uses the following input files.

File	Description
USERCHG	User/charge file. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
UCCP	Optional user corrective code. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
CCPB	CCP release file.
CCPT	3270 TIP release file (if BSTP=YES).
CCPD	Diagnostic release file (if DIAG=YES).
CCPR	Remote release file (if REMT=YES).
CODEPL	CCP corrective code.

CCPPH1 generates the following output files.

File	Description
PCMB	Updated combined program library.
LMFB	CCP list file. File 1 Multiplexer firmware. File 2 Dump bootstrap overlay.
LSAM	System autostart (SAM) listing (if requested).
ZMUX	CCP load module. File 1 Multiplexer firmware. File 2 Dump bootstrap overlay.
PCCP	New CCP program library, including corrective code from CODEPL.
PDGN	New diagnostic program library, including corrective code from CODEPL.
PBST	New 3270 TIP program library, including corrective code from file CODEPL.
CCPDpsrout	New diagnostic file (if DIAG=YES).
CCPRpsrout	New remote file (if REMT=YES).

File	Description
CCPTpsrout	New 3270 TIP file (if BSTP=YES).
PREM	New remote program library, including corrective code from CODEPL.
SMUX	Dump bootstrap symbol table.
ZSAM	SAM load module.
AEXP	Build parameters expand program binary.
AALK	Autolink program binary.
LIMC	Listing of Expand and Autolink programs.

Examples:

You can execute the CCPPH1 build step in several different ways, including the following:

BEGIN,CCPPH1,INSTALL,BSTP=YES,DIAG=YES,REMT=YES.

BSTP=YES, DIAG=YES, and REMT=YES signify that the 3270 TIP, online diagnostics, and the remote dump/load overlay are to be installed. This command selects the default for corrective code use.

BEGIN,CCPPH1,INSTALL,DIAG=YES.

DIAG=YES signifies the presence of the online diagnostics. This command selects the defaults for listings, remote concentrator products, and corrective code use.

BEGIN,CCPPH1,INSTALL,LIST=YES,REMT=YES.

LIST=YES automatically routes the list files to the printer. REMT=YES signifies the presence of the remote dump/load overlay. This command selects the defaults for diagnostics and corrective code use.

CCPBLB - CCP BINARY LIBRARY

The following build step generates an updated combined binary library of all CCP procedures and assembly language subroutines. Refer to General Build Step Call, earlier in this section, for descriptions of the parameters.

```
BEGIN,CCPBLB,INSTALL,LIST=option,XREF=xref,NOECS.
```

CCPBLB requires the following input files.

File	Description
USERCHG	User/charge file. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
USERBPS	User variant build parameters file. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
UCCP	Optional user corrective code. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
PCMB	Updated combined program library.
PCCP	New CCP program library including corrective code from CODEPL.
ZMUX	CCP load module. File 1 Multiplexer firmware. File 2 Dump bootstrap overlay.
AEXP	Build parameters expand program binary.
AALK	Autolink program binary.
SMUX	Dump bootstrap symbol table.
ZSAM	SAM load module.

CCPBLB produces these output files.

File	Description
CCPBpsrout	CCP release file.
BCMB	Combined CCP/diagnostics/remote binary library.
LFCA	Full compile assembly listings. File 1 Assembly source listing. File 2 Pascal source and object listing.

Example:

```
BEGIN,CCPBLB,INSTALL,LIST=YES,XREF=YES.
```

LIST=YES routes the listings generated by CCPBLB to the printer and does not make them a permanent file. XREF=YES causes the XREF program to generate a cross-reference listing of the Pascal source of CCP.

CCPVAR - CCP VARIANT

The following build step generates a CCP variant (phase 2) load module and a PICB load module based on user-supplied variant definitions on file USERBPS. Refer to General Build Step Call, earlier in this section, for descriptions of parameters.

```
BEGIN,CCPVAR,INSTALL,LIST=option,VN=vvv.
```

CCPVAR requires the following input files.

File	Description
USERCHG	User/charge file. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
UCCP	Optional user corrective code. Refer to User-Supplied Files, in CCP/Cross Permanent Files, later in this section.
PCMB	Updated combined program library.
BCMB	Combined CCP/diagnostics/remote binary library.
USERBPS	User variant build parameters file. Refer to User-Supplied Files, later in CCP/Cross Permanent Files, later in this section.
SMUX	Dump bootstrap symbol table from CCPPH1 build step.

CCPVAR generates the following output files.

File	Description
Vvvv	Variant release file.
Zvvv	CCP variant load module (vvv is the variant name).
Ivvv	PICB load module.
Svvv	Symbol table for CCP variant load module Zvvv and PICB load module Ivvv.
Lvvv	Variant load module and PICB listing.

Example:

```
BEGIN,CCPVAR,INSTALL,VN=FEP.
```

This command selects the defaults for listings. VN=FEP creates the files ZFEP, IFEP, SFEP, and LFEP (a load module, a PICB, a symbol table, and listing, respectively).

CCPEDIT - CCP LOAD MODULE FILE EDIT

The following build step patches an absolute CCPLOAD module (file named Zvvv, where vvv is the CCP variant load module) via a special MPEDIT run (refer to CYBER Cross System Build Utilities Reference Manual). The CCP build process requires this step only for those cases where there is a minor difference between an existing load module and the desired load module. Refer to General Build Step Call, earlier in this section, for descriptions of the parameters.

BEGIN,CCPEDIT,INSTALL,OLD=vvv_1,NEW=vvv_2.

CCPEDIT requires four input files.

File	Description
USERCHG	User/charge file. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
$Zvvv_1$	CCP variant load module vvv_1.
UEDZ	Optional direct or indirect access permanent file of MPEDIT directives to patch a CCP variant load module. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
$Svvv_1$	Symbol table associated with variant load module vvv_1.

CCPEDIT produces two output files.

File	Description
$Zvvv_2$	New CCP variant load module reflecting patch code.
$Svvv_2$	Copy of symbol table $Svvv_1$.

Example:

BEGIN,CCPEDIT,INSTALL,OLD=FEP,NEW=FE2.

CCPEDIT patches variant FEP with MPEDIT patch code directives from UEDZ to create a new variant FE2.

CCPLOAD - GENERATE CCPLOAD FILE

Based on user-supplied load file definitions on file USERBPS, the following build step generates a CCP load file used by network access method/network supervisor (NAM/NS) to downline load network processor units (NPUs). Refer to General Build Step Call, earlier in this section, for a description of the parameter.

BEGIN,CCPLOAD,INSTALL,GN=zzz.

CCPLOAD requires these input files.

File	Description
PCMB	Updated combined program library. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
USERCHG	User/charge file. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
UCCP	Optional user corrective code. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
USERBPS	CCP load file definitions file supplied by user. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
ZMUX	MUX firmware and dump bootstrap load module.
ZSAM	System autostart load module.
Zvvv	Previously created CCP variant load module(s).
Ivvv	Previously created PICB load module(s).

CCPLOAD generates one output file.

File	Description
Gzzz	CCP load file (zzz is the value associated with the GN keyword).

Example:

BEGIN,CCPLOAD,INSTALL,GN=XYZ.

This command uses the load file definitions on file USERBPS and creates the CCP load file GXYZ used by NAM/NS.

Relocate file:

If the released version of the NS job skeleton JOBNS is used (refer to NAM5 - Network Access Method Version 1 in this section), rename Gzzz file as NLFFILE and move NLFFILE to NETOPS (user index 377772_8).

CCPPURG - CCP/CROSS INSTALLATION FILES PURGE

The following optional build step purges all noncritical permanent files created by the CCP/Cross installation process. This step does not purge the user-supplied files and the CCP load file created by CCPLOAD. Refer to General Build Step Call, earlier in this section, for descriptions of parameters.

BEGIN,CCPPURG,INSTALL,V1=vvv$_1$,V2=vvv$_2$,...,Vn=vvv$_n$.

CCPPURG requires one input file.

File	Description
USERCHG	User/charge file. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.

This step produces no output files.

Example:

BEGIN,CCPPURG,INSTALL,V1=FEP,V2=LOC,V3=REM,V4=NP1,V5=NP2.

This command purges all permanent files associated with the variants FEP, LOC, REM, NP1, and NP2. A CATLIST after this procedure verifies that the system has purged all the permanent files created by the installation process. CCPPURG does not purge the files USERCHG, USERBPS, UCCP, UCRS, UEDZ, and the load file (Gzzz) created by CCPLOAD.

CCP/CROSS PERMANENT FILES

All permanent files generated by the CCP/Cross installation procedures are named by the following convention: each name consists of 4 characters and the first character identifies the file type. The first character can be any of the following file types:

File Type	Description
A	Absolute load file (Cross program).
B	Binary library or LGO file.
C	Permanent corrective code in Update format with master control character of /.
G	CCP load file created by the load file generator (LFG) program.
I	Program initiation control block (PICB)
L	CCP/Cross listing (generated during installation).
P	Program library in Update format.
S	CCP symbol table.
U	User supplied corrective code file.
Z	Load modules required by CCPLOAD.

An alphabetical list of permanent files generated by the CCP/Cross installation follows. Files are grouped by their file types.

Absolute Load Files	Description
AALK	Autolink program.
AASM	Cross macro assembler.
ACYP	CYBER 180, CYBER 170, CYBER 70, and 6000 Computer Systems Pascal compiler.
AEDT	MPEDIT program.
AEXP	Build parameters expand program.
AFMT	Pascal binary output formatter program.
ALIB	MPLIB program.
ALNK	MPLINK program.
AMAC	Macro assembler text file.
AMAS	Cross micro assembler.
APAS	MP17 Pascal compiler.
AXRF	Pascal cross-reference program.

Binary Library File	Description
BCMB	Combined CCP/diagnostics/remote binary library.

Corrective Code File	Description
CODEPL	Corrective code for CROSS/CCP/Network Host Products.

CCP Load File	Description
Gzzz	CCP load file generated by CCPLOAD (the zzz appended to the letter G is the value of the GN parameter).

CCP/PICB Load Modules	Description
Ivvv	CCP program initiation control block load modules.

CCP/Cross Listings	Description
LCRB	Cross system listings.
LFCA	Full compile assembly listings.
LIMC	Expand and autolink program listings.
LMFB	MUX firmware and dump bootstrap overlay listings.
Lvvv	Variant load module listing (vvv is variant name).

Program Libraries	Description
PBST	New 3270 TIP program library including corrective code on CODEPL.
PCCP	New CCP program library including corrective code on CODEPL.
PCMB	Updated combined program library.
PCRS	New Cross program library including corrective code on CODEPL.
PDGN	New diagnostic program library including corrective code on CODEPL.
PREM	New remote program library including corrective code on CODEPL.

Symbol Tables	Description
SMUX	Symbol table for dump bootstrap.
Svvv	Symbol table for load module Zvvv.

NOTE

All CCP/Cross user-supplied files must be permanent files under the same user name used for the build step jobs. The USERBPS and USERCHG files must be indirect access permanent files. The UCRS, UCCP, and UEDZ files can be indirect or direct access permanent files; local files of the same name are ignored.

User-Supplied Files	Description
UCCP	Optional direct or indirect access permanent file of user code corrections to CCP. The contents of this file should be the same for all build steps requiring it.
UCRS	Optional direct or indirect access permanent file of user code corrections to Cross. This file may be used only with build step CROSS.
UEDZ	Optional direct or indirect access permanent file of MPEDIT directives to patch a CCP variant load module. This file may be used only with build step CCPEDIT.
USERBPS	User build parameters file. This indirect access permanent file contains the CCP system definition, the CCP variant load module definitions, and the CCP load file definitions. This file is required for build steps CCPBLB, CCPVAR, and CCPLOAD. For each execution of CCPVAR, the USERBPS file must remain unchanged. A complete description of USERBPS immediately follows the listing of the permanent files.
USERCHG	User/charge file. This indirect access permanent file contains the USER command, the CHARGE command (if required), and commands that are executed at the start of the build step. This file is required for all build steps done from a terminal. If a CCP/Cross installation procedure is executed from a batch job, omit the USERCHG file.

Load Modules	Description
ZMUX	MUX load module firmware and dump bootstrap overlay.
ZSAM	SAM load module.
Zvvv	CCP variant load module (vvv is variant name).

NOTE

If the CCP/Cross build process is interrupted, you must ensure that the required files are present upon resumption.

USERBPS File:

Create a build parameters file (indirect access permanent file USERBPS) containing a CCP system definition, CCP variant load module definitions, and CCP load file definitions. Build steps CCPBLB, CCPVAR, and CCPLOAD require this file. During the build step CCPPH1, the utility program EXPAND searches through USERBPS for the extra program library definitions. During build step CCPBLB, the utility program EXPAND searches through USERBPS for the system (SYS) definition. It then expands the definition, according to a macro text file, into Update directives that control the options and TIPS that are assembled and compiled into the combined binary library (BCMB). For build steps CCPVAR and CCPLOAD, parameters specify the desired variant or load file definition. EXPAND then searches for and expands the definition in the same manner as described for the system definition. The Update directives created cause input to be generated for the AUTOLINK program.

USERBPS can contain any number of CCP system, variant, and load file definitions. (Refer to the sample USERBPS file at the end of this section.) If more than one system definition is present, only the first definition is used. The format of CCP build definitions follows:

$keyword_1=value_1,keyword_2=value_2,...,keyword_n=value_n.$

When a keyword takes on multiple values, the form follows:

$keyword_1=value_1/value_2/.../value_n.$

This is equivalent to the following:

$keyword_1=value_1,keyword_1=value_2,...,keyword_1=value_n.$

The following syntax rules apply to all definitions.

- The first keyword must be one that identifies the type of definition (VRD indicates a variant definition and LFD indicates a load file definition).
- EXPAND ignores all embedded blanks. Blank lines are illegal.
- A period terminates each definition.
- Continuation lines must begin with a plus (+).
- EXPAND treats any line whose first character is an asterisk (*) as a comment line.
- When a definition takes more than one line, the user should break the definition between parameter pairs.

CCP SYSTEM DEFINITION

The system definition controls the options and terminal interface programs (TIPs) that are assembled and compiled into the combined binary library (BCMB). It is similar to the variant definition (described in the following subsection), but must include all options and all TIPs that are used in any variants to be built from the resulting combined binary library.

The system definition can continue over more than one line as long as each line prior to the last ends with a comma. The last line must end with a period. The system definition has two parts, either of which may be present or absent. The resulting four formats are as follows.

Format	Significance
SYS.	No options, no TIPs.
SYS=<options>.	Options present, no TIPs.
SYS,TS=<TIPs>.	TIPs present, no options.
SYS=<options>,TS=<TIPs>.	Both options and TIPs present.

Keyword	Description
SYS=v1/v2/v3	Specifies options if present.

vi	Description
C	Support modules for CONSOLE (for printing CCP information on a terminal connected to a 2550) are compiled.
D	Online diagnostic support modules are present.
P	Support modules for statistics on line/trunk/NPU performance, which are logged on the account dayfile and the error log file, are compiled.
R	Remote concentrator products are present.
T	Support modules for TUP (test utility program) and CONSOLE are compiled. (TUP is an unsupported product.)

Keyword	Description
$TS=t_1/t_2/...$	Specifies the TIPs that are to be included in the system. TS can assume up to 10 different order-independent values.

t_i	Description
A	Asynchronous TIP is included. This TIP supports ASCII terminals, APL character sets, and IBM 2741 terminals.
B	Binary synchronous communications (BSC) TIP is included. This TIP supports the IBM 2780 and IBM 3780 terminals.
H	HASP TIP is included.

Keyword	Description

t_i	Description
M	Mode 4 TIP is included.
T	3270 TIP is included.
XP	X.25 TIP and PAD sub-TIP is included. Specify this TIP for any variant that executes in an NPU connected to a packet switching network.
XA	X.25 TIP and application-to-application sub-TIP is included.
1	User TIP1 is included.
2	User TIP2 is included.
3	User TIP3 is included.

The following example includes all the options and TIPs specified in the examples shown for the variant load module definition.

```
SYS=R/D/T,TS=A/B/H/M/XP/XA.
```

CCP VARIANT LOAD MODULE DEFINITION

The variant load module definition can continue over more than one line as long as each line ends with comma, except the last line must end with a period.

The format follows:

VRD=vvv,VT=$v_1/v_2/v_3$,SZ=xK,TS=$t_1/t_2/.../t_n$.

Keyword	Description
VRD=vvv	Identifies entry as a variant definition and specifies variant name (associated vvv value). Build step CCPVAR uses vvv to create unique permanent file names. Specify a 3-character alphanumeric string, beginning with an alphabetic character. It must not be the same as the last 3 characters of the CCP/Cross permanent file names (refer to CCP/Cross Permanent Files, earlier in this section).
VT=$v_1/v_2/v_3$	Specifies variant type of the NPU. You can associate a maximum of three separate values with VT. One of the following values must appear.

v_i	Description
F	Front-end; includes Host Interface Program (HIP) but no Local Interface Program (LIP).
L	Local; includes HIP and LIP.
R	Remote; includes LIP but no HIP.

Keyword	Description
	The following values are optional.

v_i	Description
C	Variant includes module CONSOLE.
D	Variant includes online diagnostic support modules.
P	Variant includes modules for statistics/performance results.
T	Variant includes modules TUP and CONSOLE for debugging.

Examples:

VT=L/D/T

VT=F/D

VT=R

SZ=xK	Specifies variant memory size: 65K, 81K, 96K, or 128K (x is a 2- or 3-digit number).
TS=t_1/t_2/...	Specifies which Terminal Interface Programs (TIPs) are to be included in this variant. TS can assume up to 10 different order-independent values.

t_i	Description
A	Asynchronous TIP is included. This TIP supports the ASCII terminals, APL character sets and IBM 2741 terminals.
B	Binary synchronous communications (BSC) TIP is included. This TIP supports the IBM 2780 and IBM 3780 terminals.
H	HASP TIP is included.
M	Mode 4 TIP is included.
T	3270 TIP is included.
XP	X.25 TIP and PAD sub-TIP is included. Specify this TIP for any variant that executes in an NPU connected to a packet switching network.
XA	X.25 TIP and application-to-application sub-TIP is included.
1	User TIP1 is included.
2	User TIP2 is included.
3	User TIP3 is included.

Example 1:

```
VRD=EX1,VT=L/D,SZ=81K,TS=A/M.
```

This variant supports an 81K local NPU with asynchronous and mode 4 TIPs and online diagnostics.

Example 2:

```
VRD=EX2,VT=R/C,SZ=96K,TS=A/H/XP.
```

This variant supports a 96K remote NPU with HASP, X.25 PAD, and asynchronous TIPs. This variant does not support online diagnostics but supports a 2550 console.

Example 3:

```
VRD=EX3,VT=F/D/T,SZ=128K,TS=A/B/H/M/XP/XA.
```

This variant supports a 128K front-end NPU with no remote NPUs, all TIPs (except site-defined TIPs), and online diagnostics. This variant supports a 2550 console.

CCP LOAD FILE DEFINITION

The format follows:

LFD=zzz,LM=vvv_1/vvv_2/.../vvv_n.

Keyword	Description
LFD	Identifies entry as a load file definition and specifies the last 3 characters of the load file name (associated zzz value). The zzz value must be a 3-character alphanumeric string matching the corresponding GN=zzz parameter in the build step CCPLOAD. CCPLOAD uses this value to create a unique permanent file name for the output file. zzz must not be the same as the last 3 characters of any of the CCP/Cross permanent file names (refer to CCP/Cross Permanent Files, earlier in this section).
LM	Specifies the CCP variant load modules and PICB load modules to include in this load file. The MUX firmware (phase 1), dump load, dump bootstrap, and SAM modules are automatically included in every load file. The associated value vvv_i is the 3-character name of a variant load module (file name $Zvvv_i$) that was generated by the CCPVAR build step. Repeat the vvv_i specification (separated by slants) for each variant to be included in the load file.

Example 1:

LFD=EX4,LM=EX1/EX2/EX3.

This entry defines a load file containing the variants created in the three CCP variant definition examples.

Example 2:

LFD=EX5,LM=EX3.

This entry defines a load file containing only the variant in the third CCP variant definition example.

CCP EXTRA PLS DEFINITION

The format follows:

PLS=ppp_1/ppp_2/ppp_3/.../ppp_n.

Keyword	Description
PLS	Identifies entry as an extra PL definition and specifies which user-supplied PLs should be merged into PCMB. This entry is used by the CCPPH1 step.

CCP/CROSS INSTALLATION EXAMPLES

Examples follow which illustrate installation of CCP in three network configurations: one NPU, three NPUs (two local NPUs and one remote NPU), and a multihost configuration with three hosts and two NPUs.

- All examples require the following input files.

Files	Description
USERCHG	Required for all build steps; contains USER and CHARGE commands.
UCCP	Required for user-suggested or PSR code for CCP.
USERBPS	Required for CCPBLB, CCPVAR and CCPLOAD; contains CCP system definitions, variant definitions and load file definitions.
UCRS	Required for user-suggested or PSR code for Cross.

- In the build steps, all examples use the defaults for auxiliary pack device type and inclusion/exclusion of corrective code.
- In all examples, underlined and lettered parameters indicate the interdependence among USERBPS definitions, EQPDECK entries, NDL source input, and build steps. Parameters with the same letter must match within each example.

Example 1: One NPU

The configuration includes the following.

- 81K NPU.
- Mode 4 TIP.
- Asynchronous TIP.
- No remote node support software.
- No NPU console support.
- No online diagnostics.

The following procedure illustrates the installation of CCP with a single NPU.

1. Ensure that the required files and tapes are available. Refer to figure 5-4 for appropriate USERBPS definitions, EQPDECK entries, and NDL source input.

2. Install Cross.

   BEGIN,CROSS,INSTALL.

   This step requires CRSS and a field length of 110K. Cross is needed for the CCP build process.

3. Build the phase 1 (microcode and dump bootstrap) and SAM load modules.

   BEGIN,CCPPH1,INSTALL.

   This step requires CCPB.

4. Create an updated combined binary library of all CCP Pascal procedures and assembly language subroutines.

   BEGIN,CCPBLB,INSTALL,LIST=PF,XREF=YES.

   LIST=PF stores the listings on disk as permanent files during the build job. At the end of the job, the files are copied to the new release file and purged. XREF=YES generates a Pascal cross-reference listing.

5. Create the phase 2 variant load module.

   b
   BEGIN,CCPVAR,INSTALL,LIST=PF,VN=VN1.

   The load module has a file name of ZVN1 and the PICB has a file name of IVN1. LIST=PF stores the listings on disk as permanent files during the build job. At the end of the job, the files are copied to the new release file and purged.

6. Create the load file used by NAM/NS to downline load the NPU.

   g
   BEGIN,CCPLOAD,INSTALL,GN=EX1.

   The load file name is GEX1.

```
*USERBPS
*ONE NPU EXAMPLE
*
*2550 WITH:      ASYNC TIP     NO ONLINE DIAGNOSTICS
*                MODE 4 TIP    NO REMOTE
*                81K MEMORY    NO NPU CONSOLE
*
*                                                        USERBPS
*SYSTEM DEFINITION IS                                    definition
*
*        a
SYS,TS=A/M.
*
*VARIANT DEFINITION IS
*
*    b                  a´                               a´ must be a subset of a
VRD=VN1,VT=F,SZ=81K,TS=A/M.
*
*LOAD FILE DEFINITION IS
*
*    g      b
LFD=EX1,LM=VN1.


NCF2P1:NFILE.
                              b
NP1:NPU  NODE=3, VARIANT=VN1.
                                                         NDL Source
  SUPLINK  LLNAME=LL23.                                  Input

COUP2:COUPLER  NODE=2, HNAME=HOST1.

  LL23:LOGLINK  NCNAME=NP1.
END.


                                                         EQPDECK
EQ41=NP,ST=ON,EQ=7,PI=1,CH=5,ND=2,SA=OFF.                Entry
```

Figure 5-4. USERBPS Definitions, NDL Source Input, and EQPDECK Entry for Example 1

Example 2: Three NPUs

Figure 5-5 illustrates the configuration of the network for this example. It shows the size of each NPU, the external connections (trunks, lines, and/or coupler) to each NPU, and the interface programs (TIPs and HIP and/or LIP) included in each NPU. It also shows the node number and port assignment(s) and/or NDL name for major components in the network as chosen for this example. In the configuration shown in figure 5-5:

- NPUA has three TIPs, a HIP, and a LIP. The latter two programs are required for the coupler and trunk, respectively.

- NPUB has two TIPs as well as a HIP and a LIP.

- NPUC has three TIPs and a LIP. A HIP is not required since no coupler is used.

NPUA and NPUC have online diagnostics; NPUC has console support. NPUC can communicate with the network through the remote node software of either NPUA or NPUB.

The following procedure illustrates the installation of CCP with a network configuration as shown in figure 5-5.

1. Ensure that the required files and tapes are available. Refer to figure 5-6 for appropriate USERBPS definitions, EQPDECK entries, and NDL source input.

2. Install Cross.

   ```
   BEGIN,CROSS,INSTALL.
   ```

   This step requires CRSS and a field length of 110K.

3. Build the phase 1 (microcode and dump bootstrap) and SAM load modules.

   ```
   BEGIN,CCPPH1,INSTALL,LIST=PF,DIAG=YES,REMT=YES.
   ```

   This step requires CCPB, CCPD, and CCPR. LIST=PF stores the listings on disk as permanent files; DIAG=YES specifies that online diagnostics are present; REMT=YES specifies that remote concentrator products are present.

4. Create an updated combined binary library of all CCP Pascal procedures and assembly language subroutines.

   ```
   BEGIN,CCPBLB,INSTALL,LIST=BOTH,XREF=YES.
   ```

   This step requires CCPB. LIST=BOTH routes the listings to the printer and stores them as permanent files during the build job. At the end of the job, the files are copied to the new release file and purged. XREF=YES generates a Pascal cross-reference listing.

5. Create the phase 2 variant load module for NPUA.

   ```
                              a
   BEGIN,CCPVAR,INSTALL,LIST=PF,VN=VNA.
   ```

   LIST=PF stores the listings as permanent files during the build job. At the end of the job, the files are copied to the new release file and purged. The load module file name is ZVNA, and the PICB file name is IVNA.

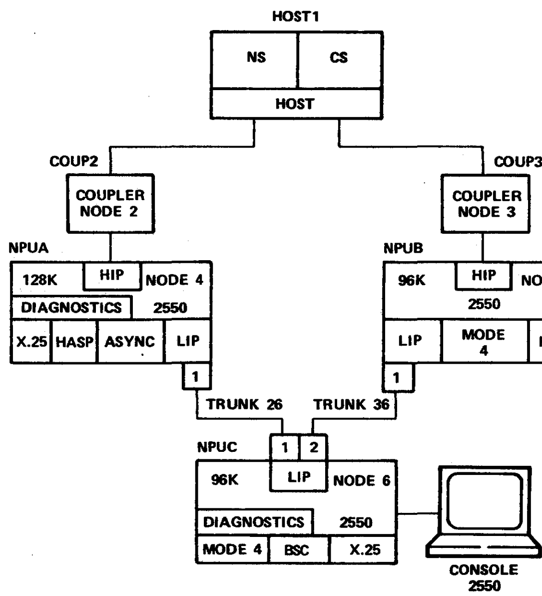

Figure 5-5. Network Configuration - Example 2

6. Create the phase 2 variant load module for NPUB.

```
                  b
BEGIN,CCPVAR,INSTALL,VN=VNB.
```

The load module file name is ZVNB and the PICB file name is IVNB.

7. Create the phase 2 variant load module for NPUC.

```
                          c
BEGIN,CCPVAR,INSTALL,LIST=PF,VN=RMC.
```

The load module file name is ZRMC and the PICB file name is IRMC. LIST=PF stores the listings as permanent files during the build job. At the end of the job, the files are copied to the new release tape and purged.

8. Create the load file used by NAM/NS to downline load the NPUs.

```
                   n
BEGIN,CCPLOAD,INSTALL,GN=EX3.
```

The load file name is GEX3.

```
*USERBPS
*THREE-NPU EXAMPLE
*
*SYSTEM DEFINITION IS
*
*   e d f    g h i j k
SYS=R/D/C,TS=A/B/H/M/XP.
*
*VARIANT DEFINITIONS ARE
*
*    a       d            g i k
VRD=VNA,VT=L/D,SZ=128K,TS=A/H/XP.
*    b                 j i
VRD=VNB,VT=L,SZ=96K,TS=M/H.
*
*    c     e d f            j h k
VRD=RMC,VT=R/D/C,SZ=96K,TS=M/B/XP.
*
*LOAD FILE DEFINITION IS
*
*    n      a   b   c
LFD=EX3,LM=VNA/VNB/RMC.
```

USERBPS Definitions

```
NCF2P1: NFILE.
                           a
NPUA: NPU NODE=4, VARIANT=VNA.

     SUPLINK LLNAME=LL24.

     COUP2: COUPLER  NODE=2,HNAME=HOST1.

     LL24: LOGLINK NCNAME=NPUA.

     LL26: LOGLINK NCNAME=NPUC.

                           b
NPUB: NPU NODE=5, VARIANT=VNB.

     SUPLINK  LLNAME=LL35.

     COUP3: COUPLER NODE=3, HNAME=HOST1.

     LL35: LOGLINK NCNAME=NPUB.

     LL36: LOGLINK NCNAME=NPUC.
```

NDL Source INPUT

Figure 5-6. USERBPS Definitions, NDL Source Input, and EQPDECK Entries for Example 2 (Sheet 1 of 2)

```
                                   c
NPUC: NPU NODE=6, VARIANT=RMC.

     SUPLINK  LLNAME=LL26.

     SUPLINK  LLNAME=LL36.

     TRUNK26:TRUNK  N1=NPUA,N2=NPUC,P1=1,P2=1.

     TRUNK36:TRUNK  N1=NPUB,N2=NPUC,P1=1,P2=2.
END.

EQ41=NP,ST=ON,EQ=7,PI=1,CH=5,ND=2,SA=OFF.                    EQPDECK
                                                             Entries
EQ42=NP,ST=ON,EQ=7,PI=1,CH=5,ND=3,SA=OFF.
```

Figure 5-6. USERBPS Definitions, NDL Source Input, and EQPDECK Entries for Example 2 (Sheet 2 of 2)

9. Execute this build step only if you want to purge extraneous files.

```
                                   a      b      c
BEGIN,CCPPURG,INSTALL,V1=VNA,V2=VNB,V3=RMC.
```

This step purges all extraneous permanent files associated with variant names VNA, VNB, and RMC.

Example 3: Three Hosts, Two NPUs, LIP trunk

Figure 5-7 illustrates the configuration of the network for this example. It shows the size of each NPU, the external connections (trunks and couplers) to each NPU, and the interface programs (TIPs, HIP, and LIP) included in each NPU. It also shows the node number and port assignments and/or NDL names for major components in the network as chosen for this example. In the configuration shown in figure 5-7 the following conditions apply:

- NPUA has four TIPs, a HIP, and a LIP. The latter two programs are required for the coupler and trunk respectively.
- NPUB has three TIPs, as well as a HIP and a LIP.
- HOST1 has NS only.
- HOST2 has CS only.
- HOST3 has CS only.
- PTF and QTF are installed on all three hosts.

NPUA has the performance statistics package and console support. NPUB has online diagnostics. NPUA and NPUB can communicate with all three hosts in the network.

The following procedure illustrates the installation of CCP with a network configuration as shown in figure 5-7.

1. Ensure that the required files and tapes are available. Refer to figure 5-8 for the appropriate USERBPS definitions, EQPDECK entries, and NDL source input.

2. Follow steps 2 through 6 in example 2.

3. Create the load file used by NAM/NS to downline load the NPUs.

```
                     n
   BEGIN,CCPLOAD,INSTALL,GN=EX4.
```

   The load file name is GEX4.

4. Execute this build step only if you want to purge extraneous files.

```
                               a       b
       BEGIN,CCPPURG,INSTALL,V1=VNA,V2=VNB.
```

   This step purges all extraneous permanent files associated with variant names VNA and VNB.

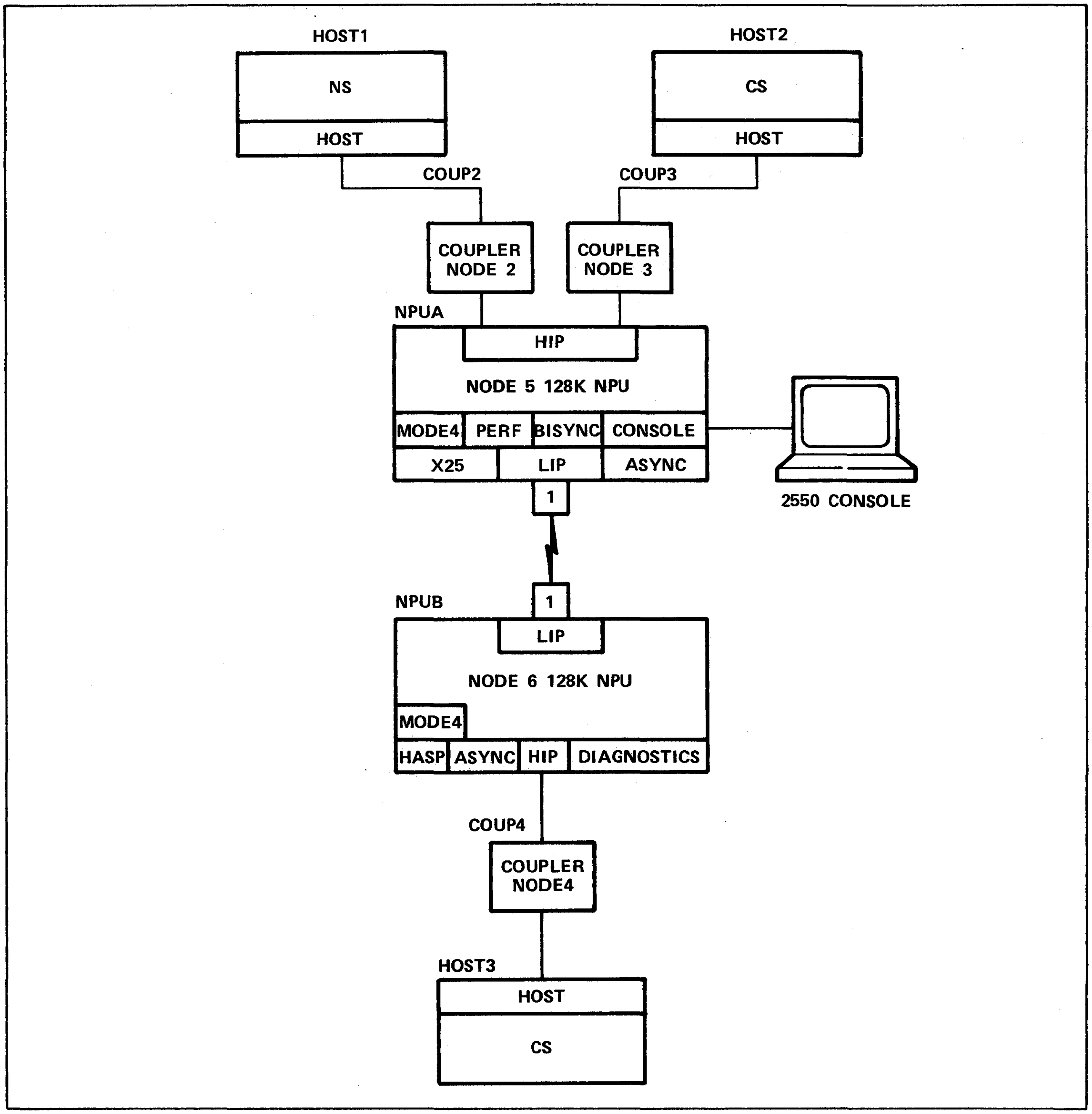

Figure 5-7. Network Configuration - Example 3

```
*   USERBPS
*
*   TWO-NPU, THREE-HOST EXAMPLE.
*
*   SYSTEM DEFINITION.
*
*   SYSTEM INCLUDES:
*
*        C       CONSOLE
*        D       DIAGNOSTICS
*        P       PERFORMANCE
*        R       REMOTE
*
*    TIPS PRESENT ARE:
*
*        A        ASYNC
*        B        BISYNC 2780-3780
*        H        HASP
*        M        MODE4
*        XA       X25 A TO A                     USERBPS
*        XP       X25 PAD                        Definitions
*
*   e d e f   g h i j k  m
SYS=C/D/P/R,TS=A/B/H/M/XA/XP.
*
*     VARIANT DEFINITIONS
*
*         NPU A
*
*    a    c b e           g h j k  m
VRD=VNA,VT=C/L/P,SZ=128K,TS=A/B/M/XA/XP.
*
*      NPU B
*
*    b    d f           g i j
VRD=VNB,VT=D/L,SZ=128K,TS=A/H/M.
*
*
*     LOAD FILE DEFINITION
*
*    n     a  b
LFD=EX4,LM=VNA/VNB.
*
*
```

Figure 5-8. USERBPS Definitions, NDL Source Input, LCF Source Input, and EQPDECK Entries for Example 3 (Sheet 1 of 5)

```
NCF2P2:NFILE.

                                  a
NPUA:NPU NODE=5, VARIANT=VNA.
  SUPLINK LLNAME=LL35.
  SUPLINK LLNAME=LL45.
  COUP2: COUPLER NODE=2, HNAME=HOST1, LOC=PRIMARY.
    LL23: LOGLINK NCNAME=COUP3.
    LL24: LOGLINK NCNAME=COUP4.
    LL25: LOGLINK NCNAME=NPUA.
    LL26: LOGLINK NCNAME=NPUB.
  COUP3: COUPLER NODE=3, HNAME=HOST2, LOC=SECOND.
    LL34: LOGLINK NCNAME=COUP4.
    LL35: LOGLINK NCNAME=NPUA.
    LL36: LOGLINK NCNAME=NPUB.

                                  b                    NDL
NPUB:NPU NODE=6, VARIANT=VNB.                          Source
    SUPLINK LLNAME=LL46.                               Input
    SUPLINK LLNAME=LL36.
    COUP4:  COUPLER NODE=4, HNAME=HOST3.
        LL45: LOGLINK NCNAME=NPUA.
        LL46: LOGLINK NCNAME=NPUB.

TRUNK56: TRUNK N1=NPUA, N2=NPUB, P1=1, P2=1.

END.



EQ41=NP,ST=ON,EQ=7,PI=1,CH=5,ND=2,SA=OFF.  COUPLER 2 FOR HOST 1
                                                                   EQPDECK
EQ42=NP,ST=ON,EQ=7,PI=1,CH=5,ND=3,SA=OFF.  COUPLER 3 FOR HOST 2      Entries

EQ42=NP,ST=ON,EQ=7,PI=1,CH=5,ND=4,SA=OFF.  COUPLER 4 FOR HOST 3
```

Figure 5-8. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 3 (Sheet 2 of 5)

```
LCFFILE:LFILE.
TITLE  LCF FOR HOST 1



**
*
*    APPLICATION DEFINITIONS FOR HOST 1
*
**


IAF      :APPL,PRIV.
RBF      :APPL,PRIV,UID,PRU.
QTF      :APPL,PRU,NETXFR,PRIV.
QTFS     :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.
PTF      :APPL,MXCOPYS=6,PRU,NETXFR,PRIV.
PTFS     :APPL,MXCOPYS=10,RS,NETXFR,PRIV.


**
*
*        INCALL/OUTCALL BLOCKS FOR HOST 1
*
**

INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=QTFS,PID=M02,SNODE=2,DNODE=3,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=QTFS,PID=M03,SNODE=2,DNODE=4,DBL=7,ABL=7,DBZ=1000.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=PTFS,PID=M02,SNODE=2,DNODE=3,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=PTFS,PID=M03,SNODE=2,DNODE=4,DBL=7,ABL=7,DBZ=1000.
```

Figure 5-8. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 3 (Sheet 3 of 5)

```
LCFFILE:LFILE.

TITLE   LCFFILE FOR HOST 2


**
*
*    APPLICATION DEFINITIONS FOR HOST 2
*
**


IAF    :APPL,PRIV.
RBF    :APPL,PRIV,UID,PRU.
QTF    :APPL,PRU,NETXFR,PRIV.
QTFS   :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.
PTF    :APPL,MXCOPYS=6,PRU,NETXFR,PRIV.
PTFS   :APPL,MXCOPYS=10,RS,PRU,NETXFRV,PRIV.

**
*
*    INCALL/OUTCALL BLOCKS FOR HOST 2
*
**


INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=QTFS,PID=MO1,SNODE=3,DNODE=2,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=QTFS,PID=MO3,SNODE=3,DNODE=4,DBL=7,ABL=7,DBZ=1000.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=PTFS,PID=MO1,SNODE=3,DNODE=2,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=PTFS,PID=MO3,SNODE=3,DNODE=4,DBL=7,ABL=7,DBZ=1000.
```

Figure 5-8. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 3 (Sheet 4 of 5)

```
LCFFILE:LFILE

TITLE   LCFFILE FOR HOST 3


**
*
*   APPLICATION DEFINITIONS FOR HOST 3
*
**


IAF    :APPL,PRIV.
RBF    :APPL,PRIV,UID,PRU.
ITF    :APPL,PRIV.
QTF    :APPL,PRU,NETXFR,PRIV.
QTFS   :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.
PTF    :APPL,MXCOPYS=6,PRU,NETXFR,PRIV.
PTFS   :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.


**
*
*    INCALL/OUTCALL BLOCKS FOR HOST 3
*
**


INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,DBL=2,ABL=2,DBZ=1000.
OUTCALL,NAME1=QTFS,PID=MO2,SNODE=4,DNODE=3,DBL=2,ABL=2,DBZ=1000.
OUTCALL,NAME1=QTFS,PID=MO1,SNODE=4,DNODE=2,DBL=2,ABL=2,DBZ=1000.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,DBL=2,ABL=2,DBZ=1000.
OUTCALL,NAME1=PTFS,PID=MO2,SNODE=4,DNODE=3,DBL=2,ABL=2,DBZ=1000.
OUTCALL,NAME1=PTFS,PID=MO1,SNODE=4,DNODE=2,DBL=2,ABL=2,DBZ=1000.
```

Figure 5-8. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 3 (Sheet 5 of 5)

Example 4: Three Hosts, Two NPUs, X25 Line

Figure 5-9 illustrates the configuration of the network for this example. It shows the size of each NPU, the external connections (trunks and couplers) to each NPU, and the interface programs (TIPs, HIP, and LIP) included in each NPU. It also shows the node number and port assignments and/or NDL names for major components in the network as chosen for this example. This configuration resembles that of example 3 except that the two NPUs are connected by an X25 line instead of a LIP trunk. In the configuration shown in figure 5-9, the following conditions apply:

- NPUA has four TIPs, a HIP, and an X25 line. The latter two programs are required for the coupler and trunk respectively.
- NPUB has three TIPs and an X25 line.
- HOST1 has NS only.
- HOST2 has CS only.
- HOST3 has NS and CS.
- PTF and QTF are installed on all three hosts.

NPUA has the performance statistics package and console support. NPUB has online diagnostics. NPUA and NPUB can communicate with all three hosts in the network.

The following procedure illustrates the installation of CCP with a network configuration as shown in figure 5-9.

1. Ensure that the required files and tapes are available. Refer to figure 5-10 for the appropriate USERBPS definitions, EQPDECK entries, and NDL source input.

2. Follow steps 2 through 6 in example 2.

3. Create the load file used by NAM/NS to downline load the NPUs.

```
                            n
   BEGIN,CCPLOAD,INSTALL,GN=EX4.
```

   The load file name is GEX4.

4. Execute this build step only if you want to purge extraneous files.

```
                            a       b
   BEGIN,CCPPURG,INSTALL,V1=VNA,V2=VNB.
```

   This step purges all extraneous permanent files associated with variant names VNA and VNB.

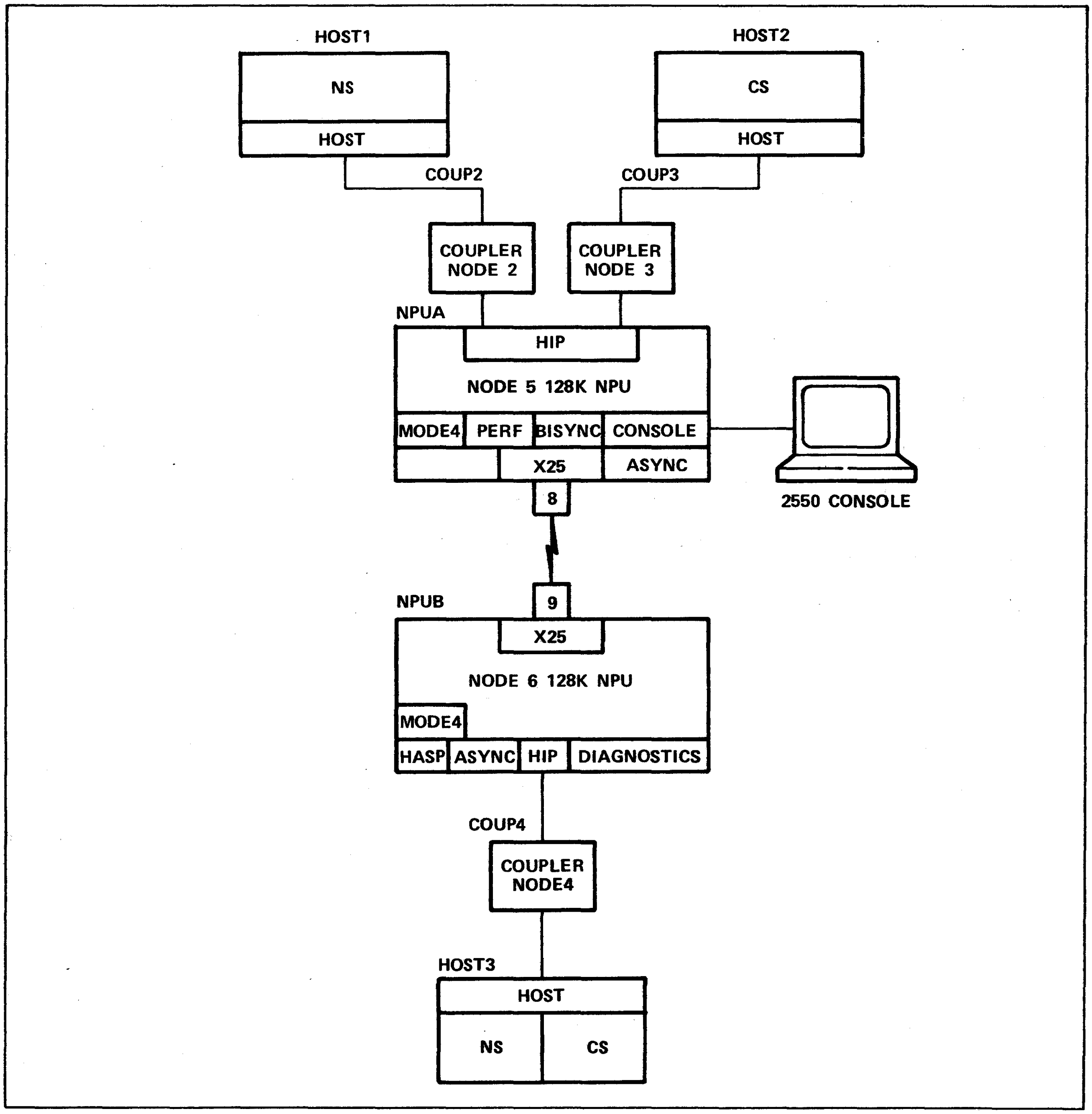

Figure 5-9. Network Configuration - Example 4

```
*   USERBPS
*
*   TWO-NPU, THREE-HOST EXAMPLE.
*
*   SYSTEM DEFINITION.
*
*   SYSTEM INCLUDES:
*
*       C       CONSOLE
*       D       DIAGNOSTICS
*       P       PERFORMANCE
*       R       REMOTE
*
*   TIPS PRESENT ARE:
*
*       A       ASYNC
*       B       BISYNC 2780-3780
*       H       HASP
*       M       MODE4
*       XA      X25 A TO A                       USERBPS
*       XP      X25 PAD                          Definitions
*
*  e d e f    g h i j k  m
SYS=C/D/P/R,TS=A/B/H/M/XA/XP.
*
*     VARIANT DEFINITIONS
*
*         NPU A
*
*    a     c b e            g h j k  m
*VRD=VNA,VT=C/F/P,SZ=128K,TS=A/B/M/XA/XP.
*
*       NPU B
*
*    b     d f            g i j k
*VRD=VNB,VT=D/F,SZ=128K,TS=A/H/M/XA.
*
*
*      LOAD FILE DEFINITION
*
*   n     a   b
LFD=EX4,LM=VNA/VNB.
*
*
```

Figure 5-10. USERBPS Definitions, NDL Source Input, LCF Source Input, and EQPDECK Entries for Example 4 (Sheet 1 of 5)

```
NCF2P2:NFILE.

                         a
NPUA:NPU NODE=5, VARIANT=VNA.
  SUPLINK LLNAME=LL35.
  SUPLINK LLNAME=LL45.
  COUP2: COUPLER NODE=2, HNAME=HOST1.
    LL23: LOGLINK NCNAME=COUP3.
    LL25: LOGLINK NCNAME=NPUA.
  COUP3: COUPLER NODE=3, HNAME=HOST2.
    LL32: LOGLINK NCNAME=COUP2.
    LL35: LOGLINK NCNAME=NPUA.
    L08   :LINE PORT=8 LTYPE=H1,
           TIPTYPE=X25,DFL=128,
           FRAME=7,RTIME=3000,RCOUNT=15,
           PSN=TYMNET,NSVC=16,DCE.
    T08   :TERMDEV W=2,NCIR=16,
           NEN=16,STIP=X25.

                         b                         NDL
NPUB:NPU NODE=6, VARIANT=VNB.                    Source
  SUPLINK LLNAME=LL46.                           Input
  SUPLINK LLNAME=LL36.
  COUP4:  COUPLER NODE=4, HNAME=HOST3.
    LL46: LOGLINK NCNAME=NPUB.
    L09   :LINE PORT=9,LTYPE=H1,
           TIPTYPE=X25,DFL=128,
           FRAME=7,RTIME=3000,
           RCOUNT=15,PSN=TYMNET,
          NSVC=16.
    T09   :TERMDEV W=2,NCIR=16,
           NEN=16,STIP=X25.

END.



EQ41=NP,ST=ON,EQ=7,PI=1,CH=5,ND=2,SA=OFF.  COUPLER 2 FOR HOST 1

EQ42=NP,ST=ON,EQ=7,PI=1,CH=5,ND=3,SA=OFF.  COUPLER 3 FOR HOST 2

EQ42=NP,ST=ON,EQ=7,PI=1,CH=5,ND=4,SA=OFF.  COUPLER 4 FOR HOST 3
```

Figure 5-10. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 4 (Sheet 2 of 5)

```
LCFFILE:LFILE.
TITLE  LCF FOR HOST 1



**
*
*   APPLICATION DEFINITIONS FOR HOST 1
*
**


IAF      :APPL,PRIV.
RBF      :APPL,PRIV,UID.
ITF      :APPL,PRIV.
QTF      :APPL,PRU,NETXFR,PRIV.
QTFS     :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.
PTF      :APPL,MXCOPYS=6,PRU,NETXFR,PRIV.
PTFS     :APPL,MXCOPYS=10,RS,NETXFR,PRIV.


**
*
*        INCALL/OUTCALL BLOCKS FOR HOST 1
*
**

INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,DBL=7,ABL-7,DBZ=1000.
OUTCALL,NAME1=QTFS,PID=M02,SNODE=2,DNODE=3,DBL=7,ABL=7,DBZ=1000.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=PTFS,PID=M02,SNODE=2,DNODE=3,DBL=7,ABL=7,DBZ=1000.
INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,PORT=8,SNODE=5,DNODE=2,DBZ=1000,DBL=7,ABL=7,
    UBZ=1000,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=QTFS,PID=M03,SNODE=2,DNODE=5,SHOST=2D3033,PORT=8,DHOST=4,DBZ=1000,DBL=7,ABL=7,
    UBZ=1000,UBL=7,WS=7,DPLS=1024.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,PORT=8,SNODE=5,DNODE=2,DBZ=1000,DBL=7,ABL=7,
    UBZ=1000,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=PTFS,PID=M03,SNODE=Z,DNODE=5,SHOST=2D3033,PORT=8,DHOST=4,DBZ=1000,DBL=7,ABL=7,
    UBZ=1000,UBL=7,WS=7,DPLS=1024.
```

Figure 5-10. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 4 (Sheet 3 of 5)

```
LCFFILE:LFILE.

TITLE   LCFFILE FOR HOST 2


**
*
*    APPLICATION DEFINITIONS FOR HOST 2
*
**


IAF     :APPL,PRIV.
RBF     :APPL,PRIV.UID.
ITF     :APPL,PRIV.
QTF     :APPL,PRU,NETXFR,PRIV.
QTFS    :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.
PTF     :APPL,MXCOPYS=6,PRU,NETXFR,PRIV.
PTFS    :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.

**
*
*    INCALL/OUTCALL BLOCKS FOR HOST 2

*
**



INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=QTFS,PID=M01,SNODE=3,DNODE=2,DBL=7,ABL=7,DBZ=1000.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,DBL=7,ABL=7,DBZ=1000.
OUTCALL,NAME1=PTFS,PID=M01,SNODE=3,DNODE=2,DBL=7,ABL=7,DBZ=1000.
INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,PORT=8,SNODE=5,DNODE=3,DBZ=1000,DBL=7,ABL=7,
    UBZ=1000,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=QTFS,PID=M03,SNODE=3,DNODE=5,SHOST=2D3033,PORT=8,DHOST=4,DBZ=1000,DBL=7,ABL=7,
    UBZ=1000,UBL=7,WS=7,DPLS=1024.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,PORT=8,SNODE=5,DNODE=3,DBZ=1000,DBL=7,ABL=7,
    UBZ=1000,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=PTFS,PID=M03,SNODE=3,DNODE=5,SHOST=2D3033,PORT=8,DHOST=4,DBZ=1000,DBL=7,ABL=7,
    UBZ=1000,UBL=7,WS=7,DPLS=1024.
```

Figure 5-10. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 4 (Sheet 4 of 5)

```
LCFFFILE:LFILE

TITLE   LCFFILE FOR HOST 3


**
*
*   APPLICATION DEFINITIONS FOR HOST 3
*
**


IAF     :APPL,PRIV.
RBF     :APPL,UID,PRIV.
ITF     :APPL,PRIV.
QTF     :APPL,PRU,NETXFR,PRIV.
QTFS    :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.
PTF     :APPL,MXCOPYS=6,PRU,NETXF2,PRIV.
PTFS    :APPL,MXCOPYS=10,RS,PRU,NETXFR,PRIV.


**
*
*    INCALL/OUTCALL BLOCKS FOR HOST 3
*
**


INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,PORT=9,SNODE=6,DNODE=4,DBZ=1000,DBL=7,ABL=7,
   UBZ=1000,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=QTFS,PID=MO1,SNODE=4,DNODE=6,SHOST=2D3031,PORT=9,DHOST=2,DBZ=1000,DBL=7,ABL=7,
   UBZ=1000,UBL=7,WS=7,DPLS=1024.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,PORT=9,SNODE=6,DNODE=4,DBZ=1000,DBL=7,ABL=7,
   UBZ=1000,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=PTFS,PID=MO1,SNODE=4,DNODE=6,SHOST=2D3031,PORT=9,DHOST=Z,DBZ=1000,DBL=7,ABL=7,
   UBZ=1000,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=QTF3,PID=MO2,SNODE=4,DNODE=6,SHOST=2D3032,PORT=9,DHOST=3,DBZ=1000,DBL=7,ABL=7,
   UBZ=1000,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=PTFS,PID=MO2,SNODE=4,DNODE=6,SHOST=2D3032,PORT=9,DHOST=3,DBZ=1000,DBL=7,ABL=7,
   UBZ=1000,UBL=7,WS=7,DPLS=1024.
```

Figure 5-10. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 4 (Sheet 5 of 5)

CDCS2 - CYBER DATABASE CONTROL SYSTEM VERSION 2

This subsection describes these installation options for CDCS2:

- Unique Parameters
- CDC Procedure File
- Special Notes
- Accounting Table
- Verification

UNIQUE PARAMETERS

SRT4 To assemble code to interface with Sort/Merge 4 rather than Sort/Merge 5, specify the keyword SRT4 on the call to CDCS2.

DEBUG To activate commands that generate CDCS flow points, specify the keyword DEBUG on the call to CDCS2. These flow points trace the execution of CDCS modules from initialization to termination. Generation of the flow points increases the execution size of CDCS by approximately 2500_8 words.

For more information on activating the interface between CDCS2 and COBOL5, refer to COBOL5 in this section.

CDC PROCEDURE FILE

Refer to the beginning of this section for information about the CDC startup procedure file. Refer to the CYBER Database Control System 2 Reference Manual for instructions on constructing the procedure file.

SPECIAL NOTES

- CDCS 2 users must have permission to use the system control point facility (refer to the description of MODVAL in the NOS 2 Administration Handbook).
- To activate a debug trace facility for CDCS 2, specify the E parameter on the SYMPL commands in the CDCS2 installation procedure.

ACCOUNTING TABLE

The CDCS routine DB$ACCT contains a table of average central processor (CP) and input/output (I/O) times, in microseconds, for CDCS user requests. These average values were obtained from simulation runs on a model 74 and adjusted based on actual runs performing file creation and updating on indexed sequential files with a record size of 40 words.

When a user issues a CDCS request, such as open, read, or rewrite, CDCS retrieves the value from the appropriate table entry and accumulates it in the accounting accumulator for the individual user. CDCS accumulates the charged CP and I/O time for all users combined and prints it in the dayfile at the end of the CDCS session. The actual time used for the entire CDCS session is also printed in the dayfile. Because different environments produce different values for the average CP and I/O times required for each user request, CDCS provides options for the database administrator to modify these table values.

One method of modification is specifying new values for the CP and I/O times on the command that initializes CDCS (refer to the CDCS 2 Data Administration Reference Manual). With this method, all the entries in the accounting table are multiplied by the ratio of the specified value to the table value for a random read on an indexed sequential file.

A second method of modification is changing the values in the accounting table and installing CDCS with the recompiled table. You can modify any or all entries in the table. List the deck DB$ACCT to see the current values in the accounting table. Entries in the table are in COMPASS macro format, as follows.

Field 1 (column 1)	Blank or comma.
Field 2 (column 2)	One of the following user request codes.

DFLOG	Logging
DFRD2	Random Read
DFRD1	Sequential Read
DFWR2	Random Write
DFSKF	Skip
DFREW	Rewrite
DFDEL	Delete
DFOPN	Open
DFCLS	Close
DFSTX	Start on index file
DFINV	Invoke
DFSTR	Start
DFEND	End
DFTER	Abnormal termination
DFRPT	Recover point
DFPVC	Privacy
DFLOK	Lock
DFULK	Unlock
DFRSR	Relation start
DFDBS	Database status block
DFRX2	Read random on index file
DFRX1	Read sequential on index file
DFRWX	Rewind index file
DFRWF	Rewind area file
DFRWR	Rewind relation
DFVER	Version change
DFBEG	Begin transaction
DFCMT	Commit transaction
DFDRP	Drop transaction
DFASK	Ask restart identifier
DFGID	Get restart identifier

Field 3 (column 11)	The macro identifier ACC.
Field 4 (column 18)	CP and I/O times required by each request. Parameters represent the different types of charges according to different file organizations, logging, and other factors. Possible parameters are as follows.

AK	AK primary key charge
ALT	Alternate key charge
ARL	Area logging flag
DA	DA primary key charge
FIX	Fix charges
ISJLG	Journal logging charge
JNL	Journal logging flag
MOD	Database modification flag
QLG	Quick recovery logging charge
QRF	Quick recovery logging flag

The following is an example of an entry in the table.

```
DFRD2  ACC  ((IS=4000,7000),(DA=3500,6500),(AK=3000,6000),(ALT=3000,7000))
```

This entry states that for a random read performed on (for example) an indexed sequential file, the CP charge is 4000 s, and the I/O charge is 7000 s.

VERIFICATION

To verify the installation of CDCS2, do the following:

1. Run a job that executes the following command.

   BEGIN,VCDCS2A,INSTALL,S=INSTALL.

   This job creates the permanent files SSIO and MSTRDIR on the installation user name INSTALL. These files are required for CDCS2 operation during verification.

2. Enter the following DSD commands at the system console.

   SUBSYST.
   L.ENABLE,CDC,n.
   L.END
   CDC.

   n is the control point.

3. Run a job that executes the following command.

   BEGIN,VCDCS2B,INSTALL.

   This job creates the permanent file IOAREAB. If this job is successful, CDCS2 verification is complete.

4. Access the K display with the following DSD command.

   K,CDC.

5. Enter the following command to terminate CDCS2.

   K.TERM.

CID - CYBER INTERACTIVE DEBUG VERSION 1

The following installation parameters define the size of various tables used by CID. Certain table sizes are defined by parameters in both SYMPL and COMPASS decks. If you alter such a table size, change all installation parameters defining the table size. Compile or assemble the indicated Update deck(s) to obtain sequence information.

Parameter	Default	Significance
BREAKTABSIZE TABSIZE	16 16	Number of entries in breakpoint table. Parameters are located in common decks BREAKD (SYMPL) and BREAKZ (COMPASS).
GROUPTABSIZE TABSIZE	16 16	Number of entries in group table. Parameters are located in common decks GROUPD (SYMPL) and GROUPZ (COMPASS).
TRAPTABSIZE TRAPXSIZE TABSIZE XSIZE	16 19 16 19	Number of entries in trap table. TRAPXSIZE and XSIZE must each be three greater than the table size defined by TRAPTABSIZE and TABSIZE. Parameters are located in common decks TRAPD (SYMPL) and TRAPZ (COMPASS).
ROOM54	10B	Number of words available for EACPM loader table (54 table) expansion before CID must recreate its overlays at debug time. Parameter is located in deck DBUGI.

COBOL5 AND COBOL5Q - COBOL VERSION 5

This subsection describes the following installation options for COBOL5:

- COBOL5 and COBOL5Q Installation Procedures
- Unique Parameters
- Installation Parameters

INSTALLATION PROCEDURES

COBOL Version 5 has two methods of customized installation: the full mode installation, which assembles and compiles all compiler and object library routines, and the Q mode installation (deck COBOL5Q).

Q mode installation allows you to modify only those routines affected by corrective code and/or user code (for example, activating Data Management) and local code (for example, default page size, CMU, and so forth), and to produce a new file through the COPYL utility, using the changed routines and a previous release level of COBOL5 as input. You must provide *COMPILE directives on file USER for any affected routine.

UNIQUE PARAMETERS

NOCDCS To deactivate the interface between COBOL5 and CDCS2, specify the keyword NOCDCS on the call that executes the installation procedure for COBOL5.

SRT4 To assemble code to interface with Sort/Merge 4 rather than Sort/Merge 5, specify the keyword SRT4 on the call for COBOL5.

TAF To activate the interface between COBOL5 and TAF, specify TAF on the call which executes the installation procedure for COBOL5.

INSTALLATION PARAMETERS

The COBOL5 compiler uses IPTEXT symbol definitions, which are filtered through CB5TEXT. No direct references to any IPTEXT symbols are contained in the compiler or the object routines. This allows you more flexibility in changing normal installation parameters for COBOL5.

The system obtains symbols governing machine type, character set, and CMU option from IPTEXT. To override one or more system defaults, select the desired changes from the following list and put them on file USER for the COBOL5 or COBOL5Q installation procedure.

- To change the default error termination level to T, W, F, or C, use 1, 2, 3, or 4, respectively, for level in the following statement. The DEF CB5$ET statement is in deck ASSEMOP.

  ```
  DEF CB5$ET#level#;
  ```

- To change the default organization (xx) for actual key, direct access, or indexed (IS) files from version 2 (ORG=NEW) to version 1 (ORG=OLD), locate CB5$xxOLDNEW in deck ASSEMOP and change it to:

  ```
  DEF CB5$xxOLDNEW # OLD #;
  ```

xx	Description
AK	Actual key files.
DA	Direct access files.
IS	Indexed files.

DCL - DATA CATALOGUE VERSION 2

This subsection describes product dependencies for DCL and SYSGEN functions for DCL.

PRODUCT DEPENDENCIES

The COBOL5 compiler and library must be available for the installation of DCL. The product must run from permanent files.

SYSGEN FUNCTIONS

SYSGEN(FULL) installs all files for DCL on user name LIBRARY. If you customize the DCL files, install the modified files by executing these steps:

1. Before running the DCL2 build procedure, execute these commands on your installation user name:

   ```
   PURGE(PFGDCL2)
   DEFINE(PFGDCL2/CT=S)
   RETURN(PFGDCL2)
   ```

2. Run the DCL2 build procedure. Then enter these commands at the system console:

   ```
   X.DIS.
   USER(INSTALL,INSTALL)
   ATTACH(PFGDCL2)
   SYSGEN(DCL2)
   ```

FDBF - FORTRAN DATABASE FACILITY VERSION 1

This subsection describes product dependencies for FDBF and unique parameters for FDBF.

PRODUCT DEPENDENCIES

The installation tool SYNGEN, which resides on the DDL 3 program library, must be available for the installation of FORTRAN Data Base Facility (FDBF) 1.

UNIQUE PARAMETERS

FTN4 To specify that FORTRAN 4 is the default language rather than FORTRAN 5, specify the keyword FTN4 on the call to FDBF.

FSE - FULL SCREEN EDITOR

This subsection describes the following:

- FSEEX and SMFEX Implementations
- SMF Procedure File
- Installation Parameters
- SYSGEN Functions for FSE
- SMFSTAT Procedure

FSEEX AND SMFEX IMPLEMENTATIONS

There are two implementations for the FSE: FSEEX AND SMFEX. The FSEEX program is a complete implementation of the editor and is called by the user with entry point FSE. The SMFEX program is a NOS subsystem, called by the operator, which implements a subset of the editor with performance characteristics different from FSEEX.

When the SMFEX subsystem is enabled, the operating system automatically processes a large portion of the user's FSE calls through SMFEX. When SMFEX is disabled, the FSEEX program is automatically scheduled with compatible external features. Where the NOS scheduler would process the FSEEX field length of approximately 50000_8 words, the SMFEX scheduler processes approximately 3000_8 words. SMFEX can process editing transactions most effectively if the hardware configuration provides extended memory in the amount of 3000_8 words per editing user. Extended memory can be ECS, ESM, LCM, or UEM. SMF can function without adequate extended memory, but it functions less efficiently. When SMFEX is operating, it uses a dedicated control point with a central memory (CM) field length typically ranging from 54000_8 to 70000_8 words. If the configuration provides user access to extended memory, SMFEX also uses a dedicated extended field length whose size is as much extended memory as is available, up to a limit determined by installation parameter NUMSWPECS.

In summary, the decision to enable or disable SMFEX is a tradeoff, where stand-alone FSEEX uses a large resource per user, while combined FSEEX plus SMFEX uses a large fixed resource with a small incremental resource for each user.

SMFEX is likely to improve overall performance if the hardware configuration provides at least 512K words of memory, including some form of extended memory, and if the workload includes at least 40 users simultaneously calling FSE. SMFEX is likely to degrade overall performance if the configuration provides 131K or smaller central memory, or if fewer than 20 users call FSE simultaneously.

For configurations of 262K central memory and 20 to 40 editor users, SMFEX can be expected to improve response times for those users who call the editor, but can either improve or degrade overall system performance. To evaluate the value of SMFEX, it is suggested that medium-size sites experiment by operating SMFEX on alternate days.

SMF PROCEDURE FILE

Refer to the beginning of this section for information about the SMF subsystem initiation.

Minimally, a procedure file for initiating SMFffff must include the following:

```
   .PROC,SMFffff.
   SMFEX.
1  DMB.
   SMFEX,RECOVER.
   SKIP,EXIT.
2  EXIT.
   DMB.
   DMD.
   DMD,3777777.
3  SMFEX,RECOVER.
   ENDIF,EXIT.
```

The sequence of DMB, EXIT, and SMFEX,RECOVER commands (numbered 1, 2, and 3) is mandatory for valid subsystem shutdown and abort processing.

The first SMFEX statement may be modified to be of the form:

```
SMFEX,n.
```

n is an integer from 2 to 8. This parameter determines the number of functions that can execute in parallel. The default is 3. Values of 3 or 4 are advisable for most sites. A guideline for selecting the value of n is one unit for every 25 users who simultaneously call FSE. Each increment in the value of n increases the central memory field length by approximately 4000_8 words.

The SMFffff procedure can control the amount of extended memory allocated, by preceding the first SMFEX statement with:

```
MFL,EC=m.
```

m is the extended memory field length in units of 1000_8 words.

Because SMFEX uses constant field lengths once it is fully initialized, it is advisable to select a control point number for the ENABLE command, which is either low or high. SMF then resides at one end of central memory, and has minimal effect on the storage move characteristics of the NOS job scheduler.

INSTALLATION PARAMETERS

The following parameters are defined in deck COMFSMF.

Parameter	Default	Significance
NUMSMFUSR	100	Maximum number of users that can be processed by SMFEX. NUMSMFUSR need not equal the maximum number of users who call FSE, since FSEEX is processed by the NOS scheduler when SMFEX overflows. If this parameter is increased beyond 100, the entire operating system must be reassembled with MXLF (in deck PPCOM) redefined, so that SMFEX can allocate additional negative field length for local FNT entries. NUMSMFUSR cannot be increased beyond 500.
NUMSWPECS	100	Maximum number of users that can be processed using extended memory. The extended field length is approximately 3000_8 words per user. This field length can be reduced at subsystem initiation by providing less extended memory or no extended memory in the EQPDECK. NUMSWPECS should not exceed NUMSMFUSR, and need not equal NUMSMFUSR since SMFEX will use mass storage devices for overflow.

SYSGEN FUNCTIONS FOR FSE

FSE is released with four files: FSEHELP, FSTEACH, FSEPROC, and SMFSTAT. The first three files are automatically stored under user name LIBRARY; SMFSTAT is stored under the installation user name INSTALL.

SMFSTAT PROCEDURE

FSE installation creates an indirect access permanent file on the installation user name which contains a procedure called SMFSTAT. This procedure provides some statistics for the Multiuser Full Screen Editor (SMF).

The SMF subsystem must be executing for SMFSTAT to run successfully. To use the SMFSTAT procedure, do the following:

1. Log in to IAF under the user name where SMFSTAT is stored.

2. Enter this command:

   ```
   BEGIN,,SMFSTAT.
   ```

   FSE comes up, and you can then enter the GET DATA (GD) directive. If the editor is not executing, or if the SMFSTAT procedure is executed in a noninteractive job, this directive has no effect. Otherwise, the GD directive obtains some statistic words from the editor's field length. Enter the QUIT (Q) directive to terminate this edit session. This starts up a FORTRAN program to analyze the results of the statistics obtained from the GD directive. The program displays a preliminary report on the screen, and provides a more detailed report on a local file called STATOUT.

3. Either route the STATOUT report to the printer, or view it using FSE.

FTN4, FTNTS, AND FTN5 - FORTRAN EXTENDED VERSION 4, FORTRAN EXTENDED VERSION 4 WITH INTERACTIVE OPTION, AND FORTRAN 5

This subsection describes the following:

- MODEL Parameter
- Installation Parameters for FTN4 and FTNTS
- Installation Parameters for FTN5
- Integer Multiply FCOs

MODEL PARAMETER

FTN4, FTNTS, and FTN5 reference the MODEL parameter (refer to TEXT and TEXTIO procedures in this section). Whether a computer efficiently executes the FORTRAN object code that it produced depends upon the model of the computer and the value specified in the MODEL parameter. If the value specified in the MODEL parameter is identical to the computer's model number, the object code executes efficiently. If the value specified in the MODEL parameter is different from the computer's model number, the object code executes inefficiently or not at all (refer to table 5-5).

INSTALLATION PARAMETERS FOR FTN4 AND FTNTS

Depending on the installation parameters of interest, you can obtain a listing of the parameters by assembling FTNMAC or FTNTEXT (the FTNMAC listing is much shorter) and/or FTN. FTN contains the installation paramters for the default command settings, command error processing, default file names, input/output buffer length, overlay library names, and reduce mode field length increments. The remaining parameters are in OPTIONS (called by FTNMAC/FTNTEXT).

INSTALLATION PARAMETERS FOR FTN5

Depending on the installation parameters of interest, you can obtain the parameters by assembling FTN5TEXT and FTN. FTN contains the installation parameters for default command settings, command error processing, default file names, input/output buffer length, and compiler overlay library names. The remaining parameters are in OPTIONS (called by FTN5TEXT).

Reinstall the compiler and CCG whenever you change parameters in OPTIONS. You can revise installation parameters in COMFCIP (called by FTN and INIT00) during the installation of FTN5, if you reassemble both FTN and INIT00.

INTEGER MULTIPLY FCOS

Because all code generated by the compiler assumes the existence of the integer multiply hardware option, you must install all applicable integer multiply FCOs.

Table 5-5. MODEL Parameter and FTN4, FTNTS, and FTN5 Object Code Execution

Value of MODEL Parameter in TEXT and TEXTIO	Models on which Object Code Executes Inefficiently	Models on which Object Code Does Not Execute
71, 72, 73, 171, 172, 173, 174, 720, 730	74, 175, 740, 750, 760, 810 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 875	176
74, 175	71, 72, 73, 171, 172, 173, 174, 720, 730, 740, 750, 760, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 875	176
740, 750, 760	71, 72, 73, 74, 171, 172, 173 174, 175, 720, 730, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 875	176
810, 815, 825, 830, 835, 845, 855	865 and 875. The following models apply as long as source code does not contain LEVEL 2 (direct access LCM) statements: 71, 72, 73, 74, 171, 172, 173, 174, 175, 720, 730,740, 750, 760.	176 The following models apply if the source code contains LEVEL 2 statements: 71, 72, 73, 74, 171, 172, 173, 174, 175, 720, 730, 740, 750, 760.
865, 875	810, 815, 825, 830, 835, 840, 845, 850, 855, and 860. The following models apply as long as source code does not contain LEVEL 2 (direct access LCM) statments: 71, 72, 73, 74, 171, 172, 173, 174, 175, 720, 730, 740, 750, 760.	176 The following models apply if the source code contains LEVEL 2 statements: 71, 72, 73, 74, 171, 172, 173, 174, 175, 720, 730, 740, 750, 760.
176	All models apply, except model 176, as long as source code does not contain LEVEL 2 (direct access LCM) statements.	All models apply except model 176, if the source code contains LEVEL 2 statements.

IAF - INTERACTIVE FACILITY VERSION 1

This subsection describes the following:

- Installation Parameters
- IAF Procedure File
- Special Notes

INSTALLATION PARAMETERS

The following parameters, defined in deck IAFEX, specify default values for the Application Interface Program (AIP) Trace utility in IAF.

Parameter	Default	Significance
DMCT	16200	Maximum number of messages logged before the trace file is released to the system for processing.
MXML	10	Maximum length in central memory words of a message logged on the trace file.
TJOB	TRACIAF	Micro whose string specifies the name of the procedure file containing the job commands used to process the trace file.

IAF PROCEDURE FILE

Refer to the beginning of this section for information about the IAF procedure file and subsystem initiating. IAF initiation consists of these three procedures: IAF, IAFTM, and IAFTR.

When IAFTM or IAFTR are used, they store a copy of TRACIAF under the system user index (377777_8) for subsequent use. For additional information regarding the trace utility, refer to the NOS 2 Analysis Handbook.

NOTE

The OPL deckname for the IAF procedure is IAFP.

SPECIAL NOTES

- Procedure file TRACIAF contains the commands to process the trace file. Modify procedure TRACIAF so that the USER and CHARGE commands have the appropriate parameters. The modifications can be placed on file USER.

- Two additional procedures exist to enable the console operator to select the type of trace, according to the parameter specified on the IAFEX command. In procedure IAFTM, T=* is the parameter on the IAFEX command; it causes the trace file to be processed only when IAF has terminated. In procedure IAFTR, T is the parameter on the IAFEX comand. The T parameter causes the trace file to be submitted as an input job using the TRACIAF file for the command record after every 16,200 messages have been logged on the trace file. Refer to NOS 2 Analysis Handbook for a description of the IAFEX command.

IMF1 - INFORMATION MANAGEMENT FACILITY VERSION 1

This subsection describes product dependencies and unique parameters for IMF1.

PRODUCT DEPENDENCIES

The installation tool SYNGEN, which resides on the DDL 3 program library, must be available for the installation of IMF1.

UNIQUE PARAMETERS

SRT4 — To assemble code to interface with Sort/Merge 4 rather than Sort/Merge 5, specify the keyword SRT4 on the call to IMF1.

ITF - INTERACTIVE TRANSFER FACILITY VERSION 1

This subsection describes ITF, product dependencies for ITF, the ITF command, and ITF Shutdown and Idledown.

ITF provides NOS interactive terminal users access to computer systems linked by a loosely coupled network (LCN) through the Remote Host Facility (RHF) subsystem. ITF multiplexes one or more network terminal connections through an application-to-application RHF connection to each remote host servicer application (ITFS). ITFS is provided only on the CYBER 200 VSOS operating system.

PRODUCT DEPENDENCIES

ITF serves as an application program to both the network access method (NAM) and RHF subsystems. ITF is initiated by NAMI using the NAM startup master file and remains connected to the NAM subsystem until either NAM is terminated or the NAM network operator commands IDLE or DISABLE are used. ITF remains connected to the RHF subsystem as long as there are interactive users connected to ITF.

Before you execute the ITF installation procedure, the NAM5, RHC, and RHF installation procedures must complete successfully. The ITF installation procedure modifies the NAM startup master file which is created by the NAM5 procedure under the build user index. After all products that modify this file have been installed, you can use the SYSGEN(MOVE) command to move the startup master file to the proper user name (refer to File Placement under the NAM5 procedure description in this section). The released default JOBITF record which is added to the startup master file includes the following command to call ITF.

```
ITF.
```

If you want to add any optional parameters to the ITF command (see ITF command in this section), you can either use a text editor to modify the startup master file or add Update directives to the file USER for the ITF installation procedure.

Before using ITF, you must ensure that the NDLP input for the NAM network configuration includes the following statement (refer to the Network Definition Language Version 1 Reference Manual for a complete description of this statement).

```
ITF: APPL,PRIV.
```

You must also ensure that the RCFGEN input for the RHF network configuration correctly defines all NPID, PATH, and RNAD entries for all remote hosts and includes an APPL directive for ITF similar to the following (refer to Network Configuration Statements under the RHF procedure description in this section).

```
APPL NAME=ITF,MXCONS=2,MXCOPYS=1
```

The value of the MXCONS parameter must not be less than the value of the PI parameter on the ITF command.

ITF COMMAND

Here is the format for the ITF command:

```
ITF(ML=ml,DL=dl,MA=ma,PI=pi,TE=te,CO=co)
```

Parameter	Significance
ML=ml	The mandatory logical identifier (LID) for ITF connections. If ML=ml is omitted, each user is prompted to enter a LID. If specified, ITF automatically attempts to connect each user to the specified LID. ml must be defined in the CMR LID table. Default is no LID.
DL=dl	The default LID for ITF connections. If ML=ml is omitted, dl is the default LID used by the system if the user does not enter a LID. dl must be defined in the CMR LID table. If DL=dl is omitted, ITF continues to request a LID from the user. Default is no LID.
MA=ma	The mandatory application to which users are switched when the connection terminates. ma must be 1 through 7 alphanumeric characters and can be the name of any NAM application installed in your system, or it may be an NVF command such as LOGOUT. If MA=ma is omitted, ITF prompts the user for another LID or application. If MA=LOGOUT is specified, users are logged out. Default is no application or command.
PI=pi	The maximum number of remote hosts to which ITF may simultaneously connect. pi can range from 1 through 7 and must be less than or equal to the value of the MXCONS parameter on the RHF configuration file APPL directive for ITF. Default is 2.
TE=te	The maximum number of interactive terminals that can connect to ITF. te can range from 1 through 128. Default is 128.
CO=co	The maximum number of terminals connected per remote host. co can range from 1 through 128. Default is 64.

The PI, TE, and CO parameters are constrained by the definitions of symbols MAXACN (released value = 2) and MAXTCN (released value = 64) in ITF common deck COMITBLS. The following relation exists:

$0 < \mathrm{PI} \leq \mathrm{MAXACN}$

$0 < \mathrm{CO} \leq \mathrm{MAXTCN}$

$0 < \mathrm{TE} \leq \mathrm{MIN(PI,MAXACN)*MIN(CO,MAXTCN)}$

If any parameter is not specified or is not in range, it is set to the maximum allowed.

The following space is allocated in common block COMITBLS at compile time:

(2*MAXACN) + (MAXACN*maxtcn) words

The space is allocated along with a variable number of buffers as needed to transfer data between the terminal and the remote host servicer. These tables and buffers are dynamically allocated and released under control of the Common Memory Manager.

ITF SHUTDOWN AND IDLEDOWN

When ITF is connected to RHF, do not use the IDLE,RHF to shut down RHF. If the IDLE,RHF command is entered while ITF is connected to RHF (or if RHF terminates abnormally), ITF aborts and does not restart automatically. To restart the application, use the RS parameter on the NAMI command.

Here are three ways to initiate an orderly shutdown of ITF:

- RHF K Display command K.IDLE

- K.DISABLE,n command

  If the K.IDLE command fails to terminate RHF within a reasonable time, check the RHF application display (K.APPL). If the active column shows that ITF is still connected, enter the K.DISABLE,n command to disable the ITF application ordinal.

- DROP,jsn command

  If ITF remains connected after you have issued the K.DISABLE,n command, determine the jsn of the ITF job and use the DROP,jsn command to restart ITF. (The ITF jsn can be determined from the NAM status K-display.)

Once ITF is no longer connected to RHF, you can use the IDLE or STOP commands to abort the subsystem. In general, you can use the DROP command to restart ITF. The default NAMI JOBITF record allows for a maximum of 10 restarts and saves ITF dump and trace files on permanent files for retrieval by COLLECT when NAM is next reinitiated.

To temporarily disable ITF connections, use these commands:

RHF APPL K.DISABLE,n

and

K.ENABLE,n

To terminate ITF operation, use these commands:

NAM NVF DISABLE,AP=ITF

and

IDLE,AP=ITF

To control access to particular LIDs and/or PIDs, use these commands:

RHF PATH

or

ID K.DISABLE,n

and

K.ENABLE,n

LCS3 AND FCS3 - CONVERSION AIDS SYSTEM VERSION 3

The Conversion Aids System includes the Language Conversion Aids System (LCS) and the File Conversion Aids System (FCS). This subsection describes USER file directives for LCS3.

USER FILE DIRECTIVES FOR LCS3

The tables of the FORTRAN and COBOL language conversion processors (LCPs) may overflow when programs with large numbers of symbols or with lengthy statements are processed.

The FORTRAN LCP name table contains a fixed-size entry for each name that appears in a declarative statement. The COBOL LCP name table contains a variable-size entry for each special name, file name, and data name, except within either an RD entry in the Report Section or an SD entry in the File Section. COBOL name table entries are 4+(n+9)/10 words long (n is the number of characters in the name), with no rounding up. For example, if n=4, the name table entry is 5 words; if n=20, the name table entry is 6 words.

You can enlarge these tables by including either of the following Update directives on file USER in the installation job for LCS.

```
*DEFINE LTAB
*DEFINE LTAB,XLTAB
```

Table sizes and central memory requirements are shown in table 5-6.

Table 5-6. Table Sizes and Central Memory Requirements

No *DEFINE LCP Name Table	(Default)	*DEFINE LTAB	*DEFINE LTAB,XLTAB
FORTRAN			
Table size	300 entries	600 entries	-
Minimum central memory required	61000_8 words	65000_8 words	-
COBOL			
Table size	3200 words	7000 words	13000 words
Minimum central memory required	60000_8 words	70000_8 words	106000_8 words

To create a special version of the LCS that includes the COPY processing logic and additional CRM routines, make the following Update directive available on file USER when the job to install the LCS is run.

```
*DEFINE CBLCOPY
```

The central memory requirements for this version of the LCS are increased by approximately 20400_8 words.

To create a special version of the LCS that generates COBOL sequence numbers in increments of 1 (the default is 10), make the following Update directive available on file USER when the job to install the LCS is run.

```
*DEFINE COLUMN6
```

The central memory requirements for this version are increased by 5 words.

LOADER - CYBER LOADER VERSION 1

This subsection describes the unique parameters for the LOADER installation procedure and the installation parameters for LOADER.

UNIQUE PARAMETERS

PRESET

The PRESET parameter sets the default central memory presetting options. The default value for the binaries you receive depends on the information you provided in the Order Information Package (OIP). The default for this procedure is zero. You can enter ZERO (same as 1), DEBUG (same as 11), or ONES (same as 2). You can also enter the values listed below. (Note that xx xxxx is the address of the preset location; yy yyyy is the result of adding 400000_8 and the address of the preset location.)

Value	Preset (Octal)				
0	No presetting for ECS; same as 1 for CM.				
1	0000	0000	0000	0000	0000
2	7777	7777	7777	7777	7777
3	1777	0000	0000	0000	0000
4	3777	0000	0000	0000	0000
5	6000	0000	0000	0000	0000
6	4000	0000	0000	00xx	xxxx
7	2525	2525	2525	2525	2525
10	5252	5252	5252	5252	5252
11	6000	0000	0004	00yy	yyyy

MAP

This parameter sets the default LOADER MAP option. The default is OFF. You can enter the following values.

Value	Description
OFF or 0	Specifies MAP(OFF): no map.
PART or 3	Specifies MAP(PART): statistics and block map.
ON or 13B	Specifies MAP(ON): statistics, block map, and entry point cross-references.
FULL or 17B	Specifies MAP(FULL): statistics, block map, entry point map, and entry point cross-references.

FLMSG This parameter determines if LOADER issues a dayfile message indicating the field length required for loading and execution. Allowable values are YES (a dayfile message is issued) and NO (loader applies for relocatable loads when no map is specified).

NOTE

Do not add code to set the symbols IP.PSET, IP.MAP, or IP.FLMSG, because these are set by the above parameters.

INSTALLATION PARAMETERS

You can change the following parameters for LOADER. Insert the parameter changes at LDRCOM.13 in the update.

Parameter	Default	Significance
IP.PSET	0	Central memory presetting options. The default value for this parameter depends on the information you provided in the OIP. (Note that xx xxxx is the address of the preset location; yy yyyy is the result of adding 400000_8 and the address of the preset location.)

Value	Preset (Octal)				
0	No presetting for ECS; same as 1 for CM.				
1	0000	0000	0000	0000	0000
2	7777	7777	7777	7777	7777
3	1777	0000	0000	0000	0000
4	3777	0000	0000	0000	0000
5	6000	0000	0000	0000	0000
6	4000	0000	0000	00xx	xxxx
7	2525	2525	2525	2525	2525
10	5252	5252	5252	5252	5252
11	6000	0000	0004	00yy	yyyy

Parameter	Default	Significance
IP.REW	1	Specifies whether file is rewound prior to beginning of load; one of the following values.

Value	Description
1	File is rewound.
0	File is not rewound.

Parameter	Default	Significance
IP.LDER	1	Error processing by the loader; one of the following values.

Value	Description
0	Abort on all errors (ERR=ALL).
1	Abort on fatal errors (ERR=FATAL).
2	No abort if abort is possible (ERR=NONE).

Parameter	Default	Significance
IP.FLINC	4000B	Amount of field length increase if additional field length is required for table construction by LOADER. Acceptable values are multiples of 100_8.

Parameter	Default	Significance
IP.LRT	0	If nonzero, a message giving various time and memory measurements is issued to the dayfile.
IP.LDBG	0	If nonzero, conditional code to aid in debugging the loader is assembled.
IP.FLMSG	1	If nonzero, a dayfile message giving field length required for loading and execution is issued; applies for relocatable loads when no map is specified. This parameter is controlled by the unique parameter FLMSG on the call to the LOADER installation procedure.
IP.MAP	0	Default loader map option. This parameter is controlled by the unique parameter MAP on the call to the LOADER installation procedure. You can enter one of the following values:

Value	Description
0	Specifies MAP(OFF); no map.
3	Specifies MAP(PART); statistics and block map.
13B	Specifies MAP(ON); statistics, block map, and entry point cross-references.
17B	Specifies MAP(FULL); statistics, block map, entry point map, and entry point cross-references.

LOADER also uses the symbol IP.MECS, which is defined in IPARAMS during the installation of TEXT and TEXTIO.

MAS - MASS STORAGE ARCHIVAL SUBSYSTEM VERSION 1

This subsection describes the following installation options for MAS:

- Unique Parameters
- MAS Procedure File
- MAS Configuration File

UNIQUE PARAMETERS

SAVELIB If you want to save M86LIB as a direct access file, specify the keyword SAVELIB on the call to install MAS.

MAS PROCEDURE FILE

Refer to Subsystem Initiation, at the beginning of this section for more information about the MAS procedure file and about subsystem initiation. The OPL deckname for the MAS procedure is MAS.

MAS CONFIGURATION FILE

You must create an MAS configuration file using the SSBLD statements. Refer to the NOS 2 Analysis Handbook for information about creating the MAS configuration file. SYSGEN(FULL) installs a sample MAS configuration file (named MSASCON) on user name SUBFAM0 (user index 377760 octal).

MCS - MESSAGE CONTROL SYSTEM VERSION 1

This subsection describes the following:

- Unique Parameters
- Installation Parameters
- MCS Procedure File
- Special Notes
- Verification

UNIQUE PARAMETERS

TRACE — If you want to activate the MCS debug facility, specify the keyword TRACE on the call to install MCS.

INSTALLATION PARAMETERS

The following parameters are defined in common deck IPA$MCS. To change these parameters, place the appropriate Update directives on file USER for the MCS installation job.

Parameter	Default	Significance
MAXFL	110000	Maximum field length (octal) to which MCS can expand.
OUTLIMIT	60	Number of messages that can accumulate in an output queue before SEND requests are rejected.

The following parameters assign relative weights to the various requests that a COBOL program can make to MCS. When the program disconnects from MCS, the accounting routine adds the corresponding weight factors of all requests and enters the total into the system account file.

Parameter	Default Value	COBOL Request
AC$ACCEPT	1	Accept.
AC$CHECKPT	1	Check point.
AC$DISABLE	1	Disable.
AC$ENABLE	1	Enable.
AC$INITIAL	2	Initial.
AC$PURGE	2	Purge.
AC$RECEIVE	3	Receive.
AC$SEND	3	Send.
AC$STOPRUN	2	Stop run.

MCS PROCEDURE FILE

Refer to the beginning of this section for information about the MCS procedure file and about subsystem initiation.

NOTE

When MCS is started by the NAMI master startup file, the ENABLE and DSD AUTO commands are not applicable to MCS.

Parameters in the procedure control the following aspects of MCS initialization.

- Default user name and family.
- Default Application Definition Language (ADL) file name.
- Operator interaction.

The default user name for MCS is SYSTEMX. To change the user name, insert a USER command in the procedure that specifies the user name and family under which MCS is to run.

To change the default ADL file, include an ATTACH or GET command in the procedure so that the local file name ADLLIB exists before MCS is called. If file ADLLIB does not exist locally, MCS tries to acquire a file with the name ADLLIB under either the default user name or the name specified with the USER command.

Inclusion of a GO parameter on the MCS program call command prevents operator interaction during MCS initialization.

The released MCS startup procedure attaches ADLLIB from the installation user name INSTALL and starts MCS automatically.

Consider the following two procedures.

Example 1:

```
.PROC,MCS.
RETURN,MCS.
RFL,60000.
MCS,GO.
EXIT.
REWIND,ZZZZZDN.
DLFP,I=0.
```

Example 2:

```
.PROC,MCSTEST.
USER,username,password,familyname.
RFL,60000.
ATTACH,ADLLIB/UN=username.
MCS.
EXIT.
REWIND,ZZZZZDN.
DLFP,I=0.
```

Example 1, the procedure named MCS, specifies the default user name (SYSTEMX) and the default ADL file (ADLLIB); it does not allow the operator to change initialization parameters.

Example 2, the procedure named MCSTEST, specifies a different user name, family name, and ADL file and allows the operator to change initialization parameters. The call to DLFP is required only if you use a debug version of MCS.

SPECIAL NOTES

- To activate debug dumps for the ADL processor, include a *DEFINE DEBUG directive on file USER for the MCS installation job.
- Users must be validated to access MCS (refer to MODVAL in the NOS 2 Administration Handbook).

VERIFICATION

The following procedure verifies the correct installation of MCS, the ADL processor, and the COBOL communications facility.

Run a job that executes the following command:

```
BEGIN,VMCS1A,INSTALL.
```

This job creates an application definition library file (ADLLIB) on the installation user name INSTALL. To start MCS processing, enter:

```
SUBSYST.
L.ENABLE,MCS,n.
L.END
MCS.
```

n is the control point number. Then, to compile a sample COBOL job and execute MCS-related verbs, run a job that executes the following command:

```
BEGIN,VMCS1B,INSTALL.
```

Optionally, you can perform a part of the verification procedure at a terminal. Bring up NAM by entering at the system console these commands:

```
SUBSYST.
L.ENABLE,NAM,n.
L.END
NAM.
```

n is the control point number. When NAM is up, log in at a terminal and specify MCS as the network application. After the MCS banner and prompt appear, enter:

```
VERIFY AOP
```

as the MCS application name and

```
TERMINAL1
```

as the symbolic name. A verification message appears at the terminal. After verification is complete, enter:

```
CFO,MCS.IDLE
```

to terminate MCS. To terminate NAM, assign the NAM K display to NVF and enter the following K display command:

```
K.DI,HO.
```

MSS - MASS STORAGE SUBSYSTEM VERSION 1

This subsection describes the following installation options for MSS:

- Unique Parameters
- MSS Procedure File

UNIQUE PARAMETERS

SAVELIB If you want to save MSSLIB as a direct access file, specify the keyword SAVELIB on the call to install MSS.

MSS PROCEDURE FILE

Refer to Subsystem Initiation, at the beginning of this section, for information about the MSS procedure file and about subsystem initiation.

MSSI - MAP SUBSYSTEM AND INTERFACE VERSION 3

There are four DECKOPL installation procedures for installing MSSI: MSSI, MMCL, AP1I, and AP1L. These procedures build the MAP subsystem, the MAP microcode to support a MAP III, MAP IV-20/21, and MAP IV-23/25, as well as online diagnostics.

INSTALLATION PROCEDURES

Here are the formats for calling the MAP installation procedures:

```
BEGIN,MSSI,INSTALL,FTN4,BLDMLIB,MSAMLIB=filename.

BEGIN,MMCL,INSTALL,FTN4.

BEGIN,AP1I,INSTALL,FTN4,MEMSIZE=size.

BEGIN,AP1L,INSTALL,FTN4,MAPTYPE=type.
```

The MSSI procedure must be run before the AP1I procedure. Refer to the MSSI Version 3 Installation Handbook for complete installation information.

MSSI VALIDATION REQUIREMENTS

The SYSGEN installation procedures for installing MSSI require the user names CDCCE and CDCCE2 (these user names can be changed, but they must be unique). The released passwords for the user names are the same as the user names. However, you can change the user names or passwords.

In the subsystem startup procedures MAPCMI, MAPECS, and MAPCH a CCL .DATA file contains the directives:

```
AP1UN=username
AP1PW-password
```

To change the user name or password, replace the parameter values username and password with the new user name and password. To use SYSGEN to load files to these user names, you must temporarily set the password and user names to the default values or you must modify SYSGEN.

PROCEDURES FOR INITIATING MSSI

CDC provides three procedure files for initiating MSSI:

MAPCMI for initiating MAP IV-23/25

MAPCH for initiating MAP IV-20/21

MAPECS for intiating MAP III

NAM5 - NETWORK ACCESS METHODS VERSION 1

This subsection describes the following installation options for NAM:

- Special Notes
- Unique Parameters
- Local and Network Configuration Files
- NAM Procedure File
- USER File Directives
- Network Startup Master File
- SYSGEN Functions

SPECIAL NOTES

- The NAM installation procedure installs the following NAM components and utilities:

  NAMI (Network Initialization Program)
  NIP (Network Interface Program)
  PIP (Peripheral Interface Program)
  NS (Network Supervisor)
  CS (Communications Supervisor)
  NVF (Network Validation Facility)
  DLFP (Debug Log File Processor)
  AIP (Application Interface Program)
  QTRM (Queued Terminal Record Manager)
  TVF (Terminal Verification Facility)
  NDLP (Network Definition Language Processor)
  LFG (CCP Load File Generator)
  NDA (NPU Dump Analyzer)
  COLLECT (Collect Network Dumps)
  LISTPPM (Format PIP Memory Dumps)

- The installation procedure retrieves the procedure and NAMSTRT files from the program library on NAM5. The installation procedure saves these files as public indirect access permanent files on the installation user name. The SYSGEN utility automatically moves the files to SYSTEMX and NETOPS. Other optional product installation procedures modify NAMSTRT (PSU, RBF, RHP with NAM interface, and ITF). Thus, you should not move NAMSTRT until these optional products have been installed.

- The NDLP binaries must be in the running system or in GLOBLIB before creating the network and local configuration files, and the LFG binaries must be in GLOBLIB before creating the CCPLOAD file. If the current NDLP binaries are not in the running system but are in GLOBLIB (they are placed there by SEED and the NAM installation procedure), the following command must be included in the job that executes NDLP. (Execution of this command must occur before NDLP is executed.)

  ```
  BEGIN,ATTGLOB,INSTALL.
  ```

- The NAM1/CCP3 Host Application Programming Reference Manual describes NIP, PIP, NS, CS, AIP, QTRM, and DLFP. The NAM/CCP Terminal Interfaces Reference Manual describes TVF. The Network Definition Language Reference Manual describes NDLP. The NOS 2 Analysis Handbook describes LFG, NAMI, COLLECT, and NDA.

- There are interdependencies between NAM and CCP. These interdependencies are established in file USERBPS (refer to CROSS and CCP in this section).

- The flow of supervisory and data messages through the network is traced by Application Interface Program (AIP) code, which creates log files of such messages. The data that the log files provide is invaluable in the analysis of error conditions in network installation or operation. Startup jobs JOBNS, JOBCS, JOBTVF, JOBNIP, and JOBNVF save the log files as direct access permanent files on tape and purges them. For network problems, this tape (with other support materials) should be included with all PSRs submitted for network products. A more detailed description of the log file capability is in the NAM1/CCP3 Host Application Programming Reference Manual.

UNIQUE PARAMETERS

NOTRACE — If you want to disable log file creation for NS, NVF, TVF, and CS, specify the keyword NOTRACE on the call to the procedure for NAM installation.

LOCAL AND NETWORK CONFIGURATION FILES

You must create the local and network configuration files with NDLP. Define these files as direct access permanent files with the names specified in the master file parameter records. LCFFILE and NCFFILE are the file names specified in the released version of the master file. Refer to the Network Definition Language Reference Manual for the procedure to create the configuration files.

LCFFILE and NCFFILE must be installed under user name NETOPS (user index 377772_8).

NAM PROCEDURE FILE

Refer to Subsystem Initiation, at the beginning of this section, for information about the NAM startup procedure file and subsystem initiation.

There are two released initiation procedure files: NAM and NAMNOGO. Use the NAM procedure to initiate NAM without operator intervention; use the NAMNOGO procedure when operator intervention is required.

A NAM procedure file causes the network initialization program NAMI to execute. NAMI takes input from a permanent file which it creates the first time it executes (under user name SYSTEMX), from its command parameters, and from the operator through CFO commands. Command parameters override the permanent file; operator entries override both the command parameters and the permanent file.

The parameters to the NAMI command primarily identify a startup master file that NAMI uses to bring up network host programs. A GO parameter on the NAMI command brings up the network without operator intervention. This parameter is provided on the NAMI command in the NAM procedure, but it is absent in the NAMNOGO procedure.

The first time the network programs are to be started, you should enter NAMNOGO to allow entry of the parameters necessary for it to execute. These parameters are the name of the startup master file, the user name and password under which the file resides, and the name of the parameter record on the master file containing additional directives to NAMI for starting the network programs. Enter the required parameters through the CFO command, ending with the GO parameter to start NAMI processing. Refer to the NOS 2 Analysis Handbook for a description of the NAMI command and for further details on NAMNOGO and the CFO command.

The released startup master file, NAMSTRT, is a multirecord file containing six parameter records (INIT, RESTRT, RECOVR, MINIT, MULTI, and MRECOV) and seven job records (JOBNIP, JOBNS, JOBCS, JOBNVF, JOBTVF, JOBCOL, and JOBPUR). For the initial NAM startup, you should specify the parameter record INIT for a single host network, or MINIT if your network contains more than one host. Subsequent entries of NAM from the console cause NAMI to use the parameter record RESTRT. If you have a multihost network, you should change the NAMI command in the NAM procedure file so that subsequent entries from the console use the parameter record MULTI.

The INIT record directs NAMI to route to the input queue jobs to start the programs NS, CS, NVF, TVF, and COLLECT. NIP is started at the NAMI control point. If RBF, ITF, QTF, and MCS have been installed, the INIT record also routes a job to start those programs.

USER FILE DIRECTIVES

To assemble the following features into NAM, include directives of the form

```
*DEFINE name
```

on file USER for the NAM5 installation procedure.

NOTE

> You should be thoroughly familiar with NAM operations before defining DEBUG and/or STAT. DEBUG and STAT are defined by the release defaults. To remove the definitions, specify NOTRACE on the call to the NAM5 installation procedure.

name	Significance When Defined
DEBUG	Code to aid in debugging and maintenance in NIP and in PIP is generated. The following shows the effect of DEBUG on NAM components. • NIP, CS, NS, and NVF abort on certain error conditions. • CS, NS, and NVF are loaded with the debug version of AIP and produce network traces. • PIP hangs PPs for certain error conditions. • NIP uses internal trace buffers to trace messages sent and received and to trace subroutine and overlay calls.
IMS	Descriptive internal maintenance comments are included in the assembly and compilation listings.
STAT	Additional statistics-producing code is generated in NIP and AIP. With STAT defined, each time an application stops talking to the network, a terminal-to-application connection terminates, or an application-to-application connection terminates, statistical information is written to the NIP dayfile. After NIP terminates, the dayfile indicates the number of times each overlay was called and gives the statistics kept in common block STATTAB. The size of the job dayfile increases significantly when STAT is defined.
ZZDN	Code is generated to log all inbound or outbound messages between PIP and NIP in local file ZZZZZDN.

The following parameter is defined in deck INPARU. To make any changes to this parameter, place the appropriate Update directive on file USER when running the NAM5 installation job.

Parameter	Default	Significance
MAXNIP	24576	Maximum field length that NIP can reach. Range is 7680 to 131071.

Use the following formula to determine a value for MAXNIP for a particular configuration. (Round MAXNIP to the nearest multiple of 64.)

$$MAXNIP = 11096 + 340h + 6560a + m + k_1w_1 + \ldots + k_nw_n + 78b_1 + 142b_2 + 206b_3$$

Variable	Description
h	Maximum number of host nodes.
a	Maximum number of applications with up through eight application-to-application connections.
m	Maximum node number.
k_iw_i	Words per terminal. This value must be determined for each of the terminals configured in the local configuration file. It depends upon both the application block limit and the network block limit defined for each terminal and the type of terminal, as follows:

Variable	Description
k_i	Application block limit (ABL) and network block limit (NBL); one of the following values.

Value	Description
1	ABL $\leq$ 2 and NBL $\leq$ 2.
ABL - 1	ABL > 2 and NBL $\leq$ 2.
NBL - 1	ABL $\leq$ 2 and NBL > 2.
ABL + NBL - 2	ABL > 2 and NBL > 2.

Variable	Description
w_i	Type of terminal; one of the following values.

Value	Description
20	Connected interactive terminal.
40	Connected batch terminal.
140	Active batch terminal.

Refer to the Network Definition Language Reference Manual for further information on ABL and NBL.

Variable	Description
b_1	Maximum number of 64-word PRU buffers (N1PRUB) that can be allocated to NAM.
b_2	Maximum number of 128-word PRU buffers (N2PRUB) that can be allocated to NAM.
b_3	Maximum number of 192-word PRU buffers (N3PRUB) that can be allocated to NAM.
	The N1PRUB, N2PRUB, and N3PRUB variables are NIP command parameters. Refer to NIP - Network Interface Program, later in this section.

NOTE

If you change the value of MAXNIP, you must also change the value of the COMSREM installation parameter NMFL. The two values should be the same.

NETWORK STARTUP MASTER FILE

The master file that NAM uses to start network processing consists of a set of text records that are either parameter records or job skeleton records. The master file is in standard NOS text format. You can modify the file using a text editor or through Update directives against the NAM PL.

A comment can follow the parameters on all statements and directives.

To list the released master file, execute the command:

```
BEGIN,NAMSLST,INSTALL.
```

TITLE Statement

The TITLE statement designates a parameter record or a job skeleton record. It must be the first statement (following the name of the record) of every record in the master file. The format follows:

```
TITLE(type)title
```

Parameter	Significance
type	Type of record; use PARAM for parameter records and JOB for job skeleton records.
title	Text string of 1 through 50 characters that you can use to describe the purpose of the record.

Example:

```
TITLE(PARAM) INIT - INITIAL NETWORK INVOCATION
```

Parameter Records

Parameter records contain two kinds of directives that tell NAMI what parameters to substitute in job skeletons (PARAM directives) and what jobs to start (JOB directives). Each record can consist of a number of lines or directives up to 80 characters long, terminated by a zero byte. Refer to the NOS 2 Analysis Handbook for a description of parameter records available with the released system.

Refer to figure 5-11 for an example of a parameter record.

```
INIT
TITLE(PARAM) INIT - INITIAL NETWORK INVOCATION.
.*
.*    THIS PARAMETER RECORD IS SELECTED WHEN THE NETWORK
.*    IS TO BE STARTED WITH FRESH LOADS OF THE FIRST
.*    LEVEL NPU(S), AND THEIR PREVIOUS CONTENTS
.*    ARE NOT TO BE DUMPED.
.*
.*
.*    1.  PURGE ALL PREVIOUS NETWORK DUMPS AND TRACES.
.*
.*    2.  LOCAL NPU(S) WILL BE STOPPED AND RELOADED
.*        WHEN THE NETWORK IS STARTED.
.*
.*    3.  LOCAL NPU(S) WILL NOT BE DUMPED WHEN THEY
.*        ARE INITIALLY LOADED.
.*
.*    4.  LOCAL NPU(S) WILL BE STOPPED IF THE HOST
.*        NETWORK FAILS.
.*
.*    5.  A FAILING NPU WILL TRIGGER HOST SUPERVISOR
.*        PROGRAM FIELD LENGTH DUMPS TO BE TAKEN AND
.*        PREVIOUS TRACE FILES TO BE PRESERVED.
.*
PARAM(NCFFN=NCFFILE) NETWORK CONFIGURATION FILE.
PARAM(LCFFN=LCFFILE) LOCAL   CONFIGURATION FILE.
PARAM(NLFFN=NLFFILE) NETWORK LOAD FILE (CCP).
PARAM(NIISTP=YES)    STOP NPU(S) AT HOST INITIALIZATION.
PARAM(NSFDP=NO)      INITIALLY LOADED NPU(S) NO DUMP.
PARAM(NIFSTP=YES)    STOP NPU(S) AT HOST FAILURE.
PARAM(NSRT=YES)      HOST DUMPS/TRACES ON NPU FAILURE.
PARAM(ZZMC=500)      MESSAGE COUNT BEFORE RELEASE TRACE FILE.
PARAM(ZZWHLCT=10)    MAX NUMBER OF AUTO RESTART OF PROGRAM.
.*
.*
JOB(JOBPUR,CO)     COLLECTOR JOB.
JOB(JOBNIP,NIP)    NAM (NIP) JOB.
JOB(JOBNVF,NV)     NVF JOB.
JOB(JOBCS,CS)      CS JOB.
JOB(JOBNS,NS)      NS JOB.
JOB(JOBTVF,TV)     TVF JOB.
.*
```

Figure 5-11. Example of Parameter Record

PARAM Directive

The PARAM directive is used in the parameter record to define keywords and values that are substituted for matching keywords in the job skeleton records. PARAM has the following format.

PARAM(keyword$_1$=value$_1$,keyword$_2$=value$_2$,...keyword$_n$=value$_n$) comment

Multiple PARAM directives can appear in a parameter record.

The following rules apply to the PARAM directive:

- Embedded spaces are ignored.
- Keywords and values can contain only letters, digits, and asterisks.
- Keywords and values must be no longer than 7 characters.
- If a keyword appears more than once, only the first definition applies.

Example:

When the following directive is present, every occurrence of the keyword NCFFN in any job skeleton record is replaced by the string NCFFILE.

```
PARAM(NCFFN=NCFFILE)  NETWORK CONFIGURATION FILE.
```

JOB Directive

The JOB directive specifies the name of a job skeleton record and a code for the name of the network product that the job skeleton starts. A JOB directive must appear in every parameter record for each Network Host Products program that NAMI should start. The JOB directive has the following format:

JOB(name,type,ssname,DI) comment

Parameter	Significance
name	Name of a job skeleton record.
type	First two or more characters of an application or program name, such as NS, CS, NIP, and so on. (Only the first 2 characters are used.)
ssname	Subsystem name if program is a subsystem (not required for NIP).
DI	If specified, the job record is not started at network initiation, but it may be started later by using the NAMI RS option when the network is operational.

The rules for format of a PARAM directive apply to the JOB directive.

Example:

```
JOB(JOBNIP,NIP)  NAM (NIP) JOB.
```

NAMI ignores all comment lines in the parameter record of the form

.*Text

Job Skeleton Records

Each job skeleton record contains the commands and input records required to start one network program. The record is in NOS job file format. In any of the commands or input record lines, any keyword (1 to 7 letters, digits, or asterisks, delimited by separators) may be a substitutable parameter. That is, if any keyword in the job skeleton record is identical to a keyword in the parameter record, NAMI substitutes the corresponding value from the parameter record for the keyword in the job skeleton. If a separator is an underscore, NAMI removes the underscore from the record when it makes the replacement.

In addition to substituting values for keywords defined in the parameter record, NAMI also substitutes variables known to it at the time it executes. These variables pertain to the startup master file currently in use and to the names of dump and trace files which NAMI creates with each new startup of NIP. Certain reserved keywords have been defined for use in the job skeleton records wherever one of the NAMI variables should be substituted.

Keyword	Meaning
CIN	Current network invocation number.
OIN	Old network invocation number.
UNM	Master file user name.
PWM	Password for master file user name.
MFN	Master file name.

The following keywords are defined by NAMI and should be used in the job skeleton record wherever the dump and trace files are to be referenced.

Keyword	Meaning
xxD0FIL	Program dumps.
xxD1FIL	
xxD2FIL	Binary dumps of field length.
xxD3FIL	
xxL0FIL	Listable output.
xxT0FIL	ZZZZZDN.
xxS0FIL	ZZZZZSN.

xx is the first 2 characters of the type from the JOB directive in the parameter record.

Job skeleton records must be constructed so that the files whose existence is assumed by the various programs are present.

NAMI ignores all comment lines in the job skeleton record of the form:

    .*text

NAMI writes an end-of-record in the submitted job when it encounters a card image of the following format in the job skeleton:

    .EOR

Refer to figure 5-12 for an example of a job skeleton record.

```
JOBNIP
TITLE(JOB)  JOBNIP - NIP RELEASE DEFAULT JOB SKELETON.
.*
.*      THIS IS THE STARTUP JOB FOR NIP.
.*
.*      THE PERMANENT FILES THAT NIP DUMPS AND TRACES ARE WRITTEN TO
.*      WILL BE COLLECTED BY THE COLLECTOR JOB ALONG WITH THE
.*      REST OF THE NETWORK TRACES AND DUMPS.
.*
.*
.*      THE FOLLOWING PARAMETERS MUST BE SET IN THE PARAMETER RECORD.
.*
.*           NIISTP = STOP NPU(S) AT HOST INITIALIZATION.
.*
.*           NIFSTP = STOP NPU(S) AT HOST TERMINATION.
.*
.*           ZZMC = MESSAGE COUNT BEFORE RELEASE OF TRACE FILE.
.*
.*
.*      PERMANENT FILES FOR RUN DATA ARE DEFINED AT JOB TERMINATION.
.*
.*      TFN          LFN          PFN                CONTENTS
.*
.*      OUTPUT       NIPOUT       NID0FIL            OUTPUT FROM JOB (DMP,DMD,ECT).
.*      ZZZZZPP      PIPDMP       NID1FIL            PIP DUMPS ON REPRIEVE.
.*      ZZZZDMB      NIPDMB       NID2FIL            BINARY FIELD LENGTH DUMPS.
.*      ZZZZTMP      NIPZTMP      NID3FIL            BINARY DUMP FILE.
.*                   NIPLST       NIL0FIL            JOB DAYFILE.
.*      ZZZZZDN      TRCLEV1                         NIP TRACE FILE WRITTEN BY NIP.
.*                   TRCLEV2      ZZNIFIL            INTERMEDIATE NIP TRACE FILE.
.*                   TRCLEV3      NIT0FIL            PERMANENT NIP TRACE FILE.
.*
.*
USER(UNM,PWM)
NORERUN.
DISPLAY(DATE)
DISPLAY(HID)
DISPLAY(OT)
DISPLAY(SC)
RETURN(OUTPUT)
SETJOB(NAM_CIN)
.*
.*      PURGE OLD LEVEL 2 TRACE FILES.
.*
PURGE(ZZNI_OIN,ZZNI_CIN/NA)
```

Figure 5-12. Example of Job Skeleton Record (Sheet 1 of 5)

```
.*
.*      START NIP.
.*
RFL(60000)
NIP(NIN=CIN,ISTP=NIISTP,FSTP=NIFSTP,MC=ZZMC)
.*
.*      NIP NORMAL TERMINATION - CHECK FOR ABNORMAL CONDITIONS.
.*
RFL(0)
SKIP(BAILOUT)
.*
EXIT.  NIP
.*
.*
.*      NIP FAILED - SAVE RUN DATA IF REQUIRED.
.*
.*
DISLAY(EF)
IF(EF.NE.SYE,BAILOUT)
SKIP(SAVFILS)
.*
ENDIF(BAILOUT)
IF(.NOT.FILE(ZZZZTMP,AS)SAVFILS)
.*
.*      NO ABNORMAL CONDITIONS - DO NOT SAVE RUN DATA ON PERMANENT FILES.
.*
PURGE(NITOFIL,ZZNI_CIN/NA)
RETURN(OUTPUT)
WRITER(OUTPUT)
EXIT.  NIP
.*
.*      SAVE RUN DATA IF AVAILABLE
.*
ENDIF(SAVFILS)
.*
IF(FILE(OUTPUT,AS),NOUTPUT)
ATTACH(NIPOUT=NIDOFIL/NA,M=W)
IF(.NOT.FILE(NIPOUT,AS))   DEFINE(NIPOUT=NIDOFIL)
REWIND(OUTPUT)
SKIPEI(NIPOUT)
COPYEI(OUTPUT,NIPOUT)
RETURN(OUTPUT,NIPOUT)
ENDIF(NOUTPUT)
.*
IF(FILE(ZZZZZPP,AS),NOZZZPP)
ATTACH(PIPDMP=NID1FIL/NA,M=W)
IF(.NOT.FILE(PIPDMP,AS))   DEFINE(PIPDMP=NID1FIL)
REWIND(ZZZZZPP)
SKIPEI(PIPDMP)
COPYEI(ZZZZZPP,PIPDMP)
RETURN(ZZZZZPP,PIPDMP)
ENDIF(NOZZZPP)
.*
IF(FILE(ZZZZDMB,AS),NOZZDMB)
```

Figure 5-12. Example of Job Skeleton Record (Sheet 2 of 5)

```
ATTACH(NIPDMB=NID2FIL/NA,M=W)
IF(.NOT.FILE(NIPDMB,AS))   DEFINE(NIPDMB=NID2FIL)
REWIND(ZZZZDMB)
SKIPEI(NIPDMB)
COPYEI(ZZZZDMB,NIPDMB)
RETURN(ZZZZDMB,NIPDMB)
ENDIF(NOZZDMB)
.*
IF(FILE(ZZZZTMP,AS),NOZZTMP)
ATTACH(NIPZTMP=NID3FIL/NA,M=W)
IF(.NOT.FILE(NIPZTMP,AS))   DEFINE(NIPZTMP=NID3FIL)
REWIND(ZZZZTMP)
SKIPEI(NIPZTMP)
COPYEI(ZZZZTMP,NIPZTMP)
RETURN(ZZZZTMP,NIPZTMP)
ENDIF(NOZZTMP)
.*
ATTACH(TRCLEV2=ZZNI_CIN/NA)
IF(FILE(TRCLEV2,AS),NTRCLV2)
SKIPR(TRCLEV2)
IF(.NOT.FILE(TRCLEV2,EOF))   REWIND(TRCLEV2)
ELSE(NTRCLV2)
IF(FILE(ZZZZZDN,AS),NOTRACE)
ENDIF(NTRCLV2)
ATTACH(TRCLEV3=NITOFIL/NA,M=W)
IF(.NOT.FILE(TRCLEV3,AS))   DEFINE(TRCLEV3=NITOFIL)
SKIPEI(TRCLEV3)
COPYEI(TRCLEV2,TRCLEV3)
PURGE(ZZNI_CIN/NA)
IF(FILE(ZZZZZDN,AS),NTRCLV1)
RENAME(TRCLEV1=ZZZZZDN)
REWIND(TRCLEV1)
IF(ZZMC.NE.0)   SKIPR(TRCLEV1)
COPYBF(TRCLEV1,TRCLEV3)
BKSP(TRCLEV3)
SKIPR(TRCLEV3)
IF(.NOT.FILE(TRCLEV3,EOF))   WRITEF(TRCLEV3)
ENDIF(NTRCLV1)
RETURN(TRCLEV1,TRCLEV2,TRCLEV3)
ENDIF(NOTRACE)
.*
ATTACH(NIPLST=NILOFIL/NA,M=W)
IF(.NOT.FILE(NIPLST,AS))   DEFINE(NIPLST=NILOFIL)
SKIPEI(NIPLST)
NOTE(DFL,NR)/NIDA_CIN
DAYFILE(DFL)
PACK(DFL)
COPEI(DFL,NIPLST)
.*
RETURN(OUTPUT)
WRITER(OUTPUT)
EXIT.  NIP
.EOR
.*
```

Figure 5-12. Example of Job Skeleton Record (Sheet 3 of 5)

```
.*
.*
.*
.*      THIS JOB IS SUBMITTED EVERY ZZMC MESSAGES TO PLACE
.*      THE TRACE INFORMATION FROM THE PROGRAM (LEVEL 1) ONTO
.*      THE INTERMEDIATE PERMANENT FILE ZZNIFIL (LEVEL 2).
.*
.*
.*      IF ALL THAT HAPPENS IS THAT THIS JOB IS REPEATEDLY
.*      SUBMITTED THEN THE TRACE INFORMATION IS KEPT FOR
.*      ONLY THE LAST 2 TIMES ZZMC MESSAGES.
.*
.*      THIS CONSTRAINS THE SIZE OF THE TRACE FILE KEPT
.*      WHEN THE NETWORK IS RUNNING WITHOUT ANY PROBLEMS
.*
.*
NIPA_CIN,T77777.    DUMP AIP TRACE TO ZZNIFIL.
USER(UNM,PWM)
ATTACH(TRCLEV2=ZZNI_CIN/M=W,NA)
IF(FILE(TRCLEV2,AS)NTRCLV2)
SKIPF(TRCLEV2)
COPYBF(TRCLEV2,TEMP)
REWIND(TRCLEV2,TEMP)
COPYBF(TEMP,TRCLEV2)
ELSE(NTRCLV2)
DEFINE(TRCLEV2=ZZNI_CIN)
WRITEF(TRCLEV2)
ENDIF(NTRCLV2)
COPYBF(INPUT,TRCLEV2)
BKSP(TRCLEV2)
SKIPR(TRCLEV2)
IF(.NOT.FILE(TRCLEV2,EOF))   WRITEF(TRCLEV2)
SETJOB(DC=NO)
EXIT.    NIPA
.EOR
.*
```

Figure 5-12. Example of Job Skeleton Record (Sheet 4 of 5)

```
.*
.*      THIS JOB IS SUBMITTED IN RESPONSE TO A
.*      *HOP RELEASE DEBUG LOGFILE* COMMAND
.*
.*      THE PURPOSE OF THIS JOB IS TO SAVE ON THE LEVEL 3 PERMANENT
.*      FILE *NITOFIL* THE PREVIOUS 2 TIMES ZZMC MESSAGES
.*      CURRENTLY IN THE INTERMEDIATE (LEVEL 2) FILE *ZZNIFIL*
.*      BEFORE WRITING THE NEW TRACE DATA (FROM LEVEL 1 FILE)
.*      ON THE INTERMEDIATE (LEVEL 2) FILE.  THIS WILL ALLOW THESE
.*      TRACE MESSAGES TO BE COLLECTED AND WRITTEN TO TAPE
.*
.*
NIPB_CIN,T77777.    DUMP TO PERMANENT TRACE FILE.
USER(UNM,PWM)
ATTACH(TRCLEV2=ZZNI_CIN/M=W,NA)
IF(FILE(TRCLEV2,AS),NTRCLV2)
SKIPR(TRCLEV2)
IF(.NOT.FILE(TRCLEV2,EOF))   REWIND(TRCLEV2)
ATTACH(TRCLEV3=NITOFIL/NA,M=W)
IF(.NOT.FILE(TRCLEV3,AS))   DEFINE(TRCLEV3-NITOFIL)
SKIPEI(TRCLEV3)
COPYEI(TRCLEV2,TRCLEV3)
EVICT(TRCLEV2)
ELSE(NTRCLV2)
DEFINE(TRCLEV2=ZZNI_CIN)
ENDIF(NTRCLV2)
WRITEF(TRCLEV2)
COPYBF(INPUT,TRCLEV2)
BKSP(TRCLEV2)
SKIPR(TRCLEV2)
IF(.NOT.FILE(TRCLEV2,EOF))   WRITEF(TRCLEV2)
SETJOB(DC=NO)
EXIT.  NIPB
```

Figure 5-12. Example of Job Skeleton Record (Sheet 5 of 5)

Network Host Product (NHP) Program Requirements

Job skeleton records for the CS, NIP, NS, and NVF jobs must each contain a command that calls the program that the job intends to run. These commands have the following format:

$prog(keyword_1=value_1,...keyword_n=value_n)$

prog	Program Name
$keyword_i=value_i$	Order-independent parameters; optional unless otherwise specified.

The files used by each program are listed following the description of each program command.

CS - Communications Supervisor

CS requires the following command.

```
CS(MC=mc,NIN=nin,CP=cputil,BU=bufutil)
```

Parameter	Description
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if mc is 0. The default value is 500.
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAMI when the network operation is started. This parameter is required.
CP=cputil	CPU utilization threshold for NPUs, from 50 through 100. The default is 100. If CPU utilization exceeds this value, the NPU operator is alerted.
BU=bufutil	Buffer utilization threshold for NPUs, from 0 through 500. The default is 0. If available buffers drop below this level, the NPU operator is notified.

CS may not be required in every host of a multihost network. If you do not want CS, remove the CS job statments from the network startup master file.

CS requires the following files.

Name	Description
NCF	Network configuration file created by NDLP.
NRF1	Job record to be copied to the ZZZZZDN file by NETREL, that determines the disposition of the network trace file, when NETREL is called on a periodic basis.
NRF2	Job record to be copied to the ZZZZZDN file by NETREL, that determines the disposition of the network trace file, when NETREL is called as a result of an operator command or NPU failure.

NOTE

The default job skeleton for CS creates NRF1 and NRF2 from the input records of the CS job.

NIP - Network Interface Program

NIP requires the following command.

```
NIP(NIN=nin,MC=mc,ISTP=istp,FSTP=fstp,N1PRUB=n1prub,N2PRUB=n2prub,N3PRUB=n3prub,INACT=ito)
```

NOTE

If a value less than the minimum is supplied, the minimum value is used.

Parameter	Description
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAMI when the network operation is started. This parameter is required.
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if mc is 0. The default value is 0.
ISTP=istp	Option to stop all local NPUs during network host initialization. The default is NO. istp / Significance YES — Stop all local NPUs. NO — Do not stop local NPUs.
FSTP=fstp	Option to stop all local NPUs upon network host termination. The default is YES. fstp / Significance YES — Stop all local NPUs. NO — Do not stop local NPUs.
INACT=ito	Inactive timeout. Time in minutes for connection to be inactive. Default is 10 minutes. After such period NIP will send FC/INACT/SM to application. If the value is zero, no FC/INACT/SM will be sent.
N1PRUB=n1prub	Maximum number of 64-word buffers that can be allocated per driver. This value is dependent on the number of batch devices and application-to-application connections whose UBZ or DBZ is specified to be 640 characters or fewer. The default value is 4. The minimum value is 2.

Parameter	Description
N2PRUB=n2prub	Maximum number of 128-word buffers that can be allocated per driver. This value is dependent on the number of batch devices and application-to-application connections whose UBZ or DBZ is specified to be 1280 characters or fewer. The default value is 8. The minimum value is 4.
N3PRUB=n3prub	Maximum number of 192-word buffers that can be allocated per driver. This value is dependent on the number of batch devices and application-to-application connections whose UBZ or DBZ is specified to be 1920 characters or fewer. The default value is 4. The minimum value is 2.

Correct PRU buffer allocation increases the throughput of the host software for batch data and significantly reduces the host software's resource use when it is handling batch traffic. The buffers specified by the values nlprub, n2prub, and n3prub are shared between all copies of PIP and all active PRU streams. The network configuration file allows you to select the number of PRUs to be transferred on connections with batch devices between the PIP and the NPU.

Since performance is related to available buffers, correct PRU buffer allocation is important. In general, a PRU buffer can support from four through six active data streams at 9600 baud. The suggested configuration is two 64-word PRU buffers (NlPRUB=2) two 128-word PRU buffers (N2PRUB=2), and two 192-word PRU buffers (N3PRUB=2) if PIP supports the PRU data transfers on all three PRU buffer sizes.

To determine if the PRU buffer allocation is correct, assemble the statistics feature (STAT) into NAM. STAT causes the NAM dayfile to display at network termination the percentage of PRU buffer use. Ideally, the batch buffer use should be between 50 percent and 70 percent. To determine if the percentage of PRU buffer use is correct at your installation, load the network and run a typical remote batch load. At the completion of the run, terminate the network. Once the PRU buffer allocation is correct, you may want to recompile NAM with STAT turned off, because the statistics feature increases the size of the job dayfile.

NOTE

Leave PIP and its associated overlays on disk. Moving these overlays to central memory does not increase performance, since PIP copies its transient overlays to NAM's field length during its initialization.

NS - Network Supervisor

NS requires the following command:

```
NS(NIN=nin,FDP=fdp,RT=rt,MC=mc,NDFCT=option)
```

Parameter	Description
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAMI when the network operation is started. This parameter is required.
FDP=fdp	Forced dump option.

fdp	Significance
YES	Within the first 10 minutes of program execution, in the absence of other NPU dumping conditions, NPUs are dumped before loading takes place.
NO	NPUs are not dumped. The default is NO.

Parameter	Description
RT=rt	Release trace file option.

rt	Significance
YES	NAM requests NS programs CS and NVF to release their trace files whenever NS dumps an NPU. The default is YES.
NO	Trace files are not requested to be released.

Parameter	Description
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if mc is 0. The default value is 500.
NDFCT=option	File category of NPU dump files. The following options can be specified.

option	Significance
PU	Public file
S	Semiprivate file.
P	Private file.

NS may not be required to run in every host of a multihost network. If you do not want NS, remove the NS job statements from the network startup master file.

NS requires the following files:

Name	Description
NLF	Network load file created by LFG in the CCPLOAD procedure (refer to the NOS 2 Analysis Handbook for a description of LFG).
NCF, NRF1, and NRF2	Described under CS, earlier in this section.

NVF - Network Validation Facility

NVF requires the following command:

```
NVF(AL=arl,LL=lrl,MC=mc,NIN=nin)
```

Parameter	Description
AL=arl	Application retry limit. This parameter specifies the maximum number of invalid application connection request attempts an application is allowed before NVF considers the application to be breaching security and aborts the application. The value can range from 1 through 4. The default is 1.
LL=lrl	Login retry limit. This parameter specifies the maximum number of invalid login attempts a user is allowed before NVF considers the user to be breaching security and terminates the connection. The value can range from 1 through 4. The default is 4.
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if mc is 0. The default value is 500.
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAMI when the network operation is started. This parameter is required.

NVF requires the following files:

Name	Description
LCF	The local configuration file created by NDLP.
NRF1 and NRF2	Described under CS, earlier in this section.

SYSGEN FUNCTIONS

Use the SYSGEN(MOVE) utility to place the files NAM and NAMNOGO on the system user index (377777_8) and to place the local and network configuration files created with NDLP, the NAMI startup master file, and the load file created by LFG under the user name NETOPS (user index 377772_8). The default file names for these files are LCFFILE, NCFFILE, NAMSTRT, and NLFFILE, respectively. The NAMI startup master file must be a public file.

SYSGEN(FULL) installs a complete NAMSTRT file to user name NETOPS with all related products (PSU, RBF, RHP with NAM interface, and ITF) merged with NAMSTRT. SYSGEN(FULL) also installs NAM and NAMNOGO to the user name SYSTEMX. SYSGEN(SOURCE) loads a copy of the standard NAMSTRT (no applications added) and a copy of the released Network Definition file (NDLDATA) to user name INSTALL. Default network configuration files (NCFFILE and LCFFILE) are installed to user name NETOPS, depending on the CCP option you selected from the OIP.

NOS AND NOS2B - NETWORK OPERATING SYSTEM

The NOS procedure assembles all COMPASS routines; the NOS2B procedure compiles the FORTRAN and SYMPL routines.

INSTALLATION PARAMETERS IN NOS DECKS

You can modify installation parameters for the operating system by following these steps:

- Execute the MODOPL procedure, incorporating any corrective code (corrective code can change the deck line numbers).
- Get a listing of the deck that contains the parameter you want to change. From the listing, get information such as line numbers, which you need to change an installation parameter.
- Put the NOS installation deck parameter changes on file USER and again execute the MODOPL procedure.

If you change any of the installation parameters in a NOS deck, reassemble all routines that use that deck. Use the KRONREF command to determine which routines use the NOS deck.

Refer to table 5-7 for brief descriptions of the NOS common decks that contain installation parameters.

Table 5-7. NOS Common Decks

Common Deck Name	A Deck That Calls the Common Deck	Description
COMEIPR	CALLMSS	Mass Storage Subsystem (MSS) parameters.
COMSACC	CALLSYS	User validation limits.
COMSBIO	CALLSYS	Central site batch I/O parameters.
COMSIOQ	CALLSYS	Dayfile/QPROTECT equivalences.
COMSJIO	CALLSYS	Devices to which users route files.
COMSLSD	CALLSYS	Search for label sector of a mass storage device.
COMSMLS	CALLSYS	Specifies micros that define multilevel security access levels and access categories.
COMSMSC	CALLSYS	Miscellaneous parameters for the operating system.
COMSMTX	CALLSYS	Magnetic tape executive routine and magnetic tape processing routine parameters.
COMSPFM	CALLSYS	Permanent file symbols and locations and formats of call blocks, catalog, and permit entries.
COMSPRO	CALLSYS	PROFILa parameters.
COMSREM	CALLSYS	Interactive Facility (IAF) parameters.
COMSRSX	CALLSYS	Job resource executive parameters.
COMSSCD	CALLSYS	Service class definitions
COMSSFS	CALLSYS	Field length limit for execution of MODVAL and PROFILE commands.
COMSSRU	CALLSYS	Parameters used in SRU calculations.
COMSSSJ	CALLSYS	Special system job parameters.
COMS1DS	CALLSYS	1DS function code definitions.
COMTNAP	CALLTAB	Valid network application parameters.
COMUSIT	CALLMSS	ASMOVE utility parameters.
COMXIPR	CALLFAS	Mass Storage Extended Subsystem (MAS) parameters.
PPCOM	PPTEXT	Maximum number of EST entries, length of L-display input and output buffers, and number of mass storage devices.

COMEIPR Parameters

COMEIPR contains the following parameters used by the MSS executive (MSSEXEC). Assemble CALLMSS to obtain a listing of COMEIPR.

Parameter	Default	Significance
NUMSLV	2	The number of slave mainframes for which the master MSSEXEC can service file staging requests.
NUMRB	9	The number of file staging request blocks available to a slave mainframe.
SLRP$INTV	5	The number of seconds that MSSEXEC waits to look for new staging requests from the slave mainframes.
SLAV$INTV	60	The number of seconds that MSSEXEC waits with no signal from a slave mainframe before assuming that the slave executive has terminated.

COMSACC Parameters

COMSACC contains a general description of the user validation file. Assemble CALLSYS to obtain a listing of COMSACC.

Parameter	Default	Significance
APFN	VALIDUS	Micro definition that specifies the name of the file containing the user names that validate user access to the operating system. Refer to the NOS 2 Analysis Handbook for further information on VALIDUS.
AUFN	VALINDS	Micro definition that specifies the name of the available user indexes file. Refer to the NOS 2 Analysis Handbook for further information on VALINDS.

The NOS 2 Analysis Handbook describes the use of the following COMSACC user control parameters.

Parameter	Default	Significance
APXL	7777_8	User password expiration term limit in days, from 1 through 7777_8. This value establishes the upper limit on the expiration term that the user can specify with the XT parameter on the PASSWOR command (refer to the NOS 2 Reference Set, Volume 3).
APXT	7777_8	Default user password expiration term in days, from 1 through 7777_8. This value is assumed when MODVAL sets a password for a new user name. APXT must be less than or equal to APXL. A value of 7777_8 indicates the password cannot expire.
KTLI	10B	Value used in calculating the default time limit; the maximum value is 176_8. For details of the algorithm used in the calculation, refer to the NOS 2 Analysis Handbook.
KLPI	1000B	Default limit for lines printed from a file; the maximum value is 3776_8.
KCPI	0	Default limit for cards punched from a file; the maximum value is 76_8.
KMSI	1000B	Default limit for additionally allocated mass storage PRUs; the maximum value is 7776_8.
KDFI	100B	Default limit for dayfile messages written; the maximum value is 176_8.
KCCI	100B	Default limit for commands processed; the maximum value is 176_8.
KECI	0	Default limit for extended memory field length/1000_8; the maximum value is 176_8.
KCMI	37B	Default limit for central memory field length/100_8; the maximum value is 37_8.
KSLI	10B	Default limit for SRU accumulation; the maximum value is 76_8.
KDTI	0	Default limit for the number of detached jobs; the maximum value is 37_8.
KPTI	1000B	Default limit for the number of units plotted; the maximum value is 76000_8.

COMSBIO Parameters

COMSBIO contains the following parameters, which are used for control of BIO functions. Assemble CALLSYS to obtain a listing of COMSBIO.

Parameter	Default	Significance
PL6L	64	Number of lines of print a user is charged for each page of output printed by BIO at six lines per inch.
PL8L	85	Number of lines of print a user is charged for each page of output printed by BIO at eight lines per inch.

COMSIOQ Parameters

COMSIOQ contains the following parameter, which is used for control of terminated dayfiles. Assemble CALLSYS to obtain a listing of COMSIOQ.

Parameter	Default	Significance
USRN	null	Micro definition that specifies the user name to which terminated dayfiles should be permitted.

COMSJIO Parameters

COMSJIO contains the following parameters, which define the devices to which the site allows users to route files. Two-character disposition codes, corresponding to the device codes defined for the ROUTE command, followed by a $ identify the legal devices. Assemble CALLSYS to obtain a listing of COMSJIO.

Parameter	Default	Significance
LP$	Defined	Any line printer.
PR$	Defined	Any line printer.
LR$	Defined	Any 580-12 printer.
LS$	Defined	Any 580-16 printer.
LT$	Defined	Any 580-20 printer.
SB$	Defined	Punch system binary.
PB$	Defined	Punch system binary.
P8$	Defined	Punch 80-column binary.
PU$	Defined	Punch coded.
PH$	Defined	Punch coded.
PL$	Defined	Plotter.

COMSLSD Parameters

COMSLSD contains the following parameter, which references information maintained in the label sector of a mass storage device. Assemble CALLSYS to obtain a listing of COMSLSD.

Parameter	Default Value	Significance
LTKL	20B	If you did not initialize a mass storage device during deadstart (using the INITIALIZE entry described in the NOS 2 Analysis Handbook), the system searches the device for a label that might be in track 0. This parameter specifies the number of tracks the system searches before determining that the device has a bad label or no label. When it reaches that track number (in the released system, track 20_8), it stops searching for a label. If the device is a system device, the system writes a new label; if it is not a system device, the error codes LE (label error) and U (unavailable) status are entered in the mass storage table (MST), and the device must be initialized after deadstart. MST is described in the NOS 2 Analysis Handbook.

COMSMLS Parameters

COMSMLS contains micros that define security access levels and access categories. Redefining any of the access level or access category micros requires reassembly of all programs that reference them. A site security administrator supplies any changes to be made to these micros. Assemble CALLSYS to obtain a listing of CCMSMLS.

Parameter	Default	Significance
ALM0 through ALM7	LVL0 through LVL7	Micro definitions that specify the names of access level 0 through access level 7.
ACM00 through ACM31	CAT00 through CAT31	Micro definitions that specify the names of access category 00 through access category 31.

COMSMSC Parameters

COMSMSC contains the following miscellaneous parameters, which are used by the operating system. Assemble CALLSYS to obtain a listing of COMSMSC.

Parameter	Default	Significance
MXSY	5	Maximum number of devices that can be defined as system devices during deadstart.
AFDL	20000B	Account file threshold size in PRUs.†
BLTL	1000B	Binary maintenance log threshold size in PRUs.†
DFDL	20000B	Dayfile threshold size in PRUs.†
ELDL	1000B	Error log threshold size in PRUs.†
UPTL	10B	Maximum number of uncorrected processor errors that can occur per hour before the operator is notified.

†When entries in any of these files reach the threshold, the A,OPERATOR flashing message appears on the console screen (refer to the NOS 2 Analysis Handbook).

COMSMTX Parameters

COMSMTX contains the following parameters, which are used by the magnetic tape executive routine and by related magnetic tape processing routines. Assemble CALLSYS to obtain a listing of COMSMTX.

Parameter	Default	Significance
MUNIT	16D	Maximum number of tape units defined per mainframe.
NTIM	300	Delay time in seconds (decimal) before reissuing the CHECK, E,P DISPLAY message at MAGNET´s control point when entries exist in MAGNET´s preview buffers.
POLM	0	Flag indicating whether all tape hardware error messages are issued to the job dayfile. If POLM is 0, the system issues only the first and last messages to the job dayfile. If POLM is 1, the system issues all tape error messages to the dayfile. The user can override the installation setting of POLM with parameters on the tape assignment command (refer to the NOS 2 Reference Set, Volume 3). The system issues all tape error messages to the error log regardless of the setting of POLM.
POGH	0	Flag indicating whether the system allows hardware-detected correctable errors when writing on 6250-cpi group-encoded (GE) tapes. The user can override the installation setting of POGH with parameters on the tape assignment command (refer to the NOS 2 Reference Set, Volume 3). If POGH is 0, the tape subsystem performs write error correction according to industry standard group-coded recording (GCR) techniques. Control Data recommends this setting because it provides efficient throughput, error recovery, and tape use when writing GE tapes on media suitable for use at 1600 cpi or 6250 cpi. If POGH is 1, hardware GCR error correction is disabled. Control Data recommends this option only for special archiving and diagnostic applications. Successful use requires higher-than-normal quality tape and special drive adjustments. Use in a normal environment generally results in increased error rates, decreased throughput, and decreased tape capacity. Use only tape that is suitable for recording at 6250 cpi when this setting of POGH is in effect. Because use of the disabled GCR error correction mode (also known as perfect write) may necessitate additional maintenance activities, consult site maintenance personnel before making this the default mode of operation.

Parameter	Default	Significance
ZFAM	A null micro	Enables conversion of binary zero family names to nonzero family names. ZFAM allows users to continue to access labeled tapes that are restricted to owner access (file accessibility field in HDR1 label is A) and were built under the binary zero family. If ZFAM is a null micro, the system default family name is substituted for the binary zero family name; otherwise, ZFAM specifies the name to be substituted.

COMSPFM Parameters

COMSPFM contains the following parameters, which are used for permanent file symbols and locations, formats of call blocks, and catalog and permit entries. Assemble CALLSYS to obtain a listing of COMSPFM.

Parameter	Default	Significance
APLO	0	Auxiliary pack load option. This parameter controls whether or not a user can request an auxiliary pack to be loaded via an MFLINK request. When APLO equals 0, MFLINK requests to auxiliary packs not currently mounted are rejected with the message DEVICE UNAVAILABLE. When APLO is equated to a nonzero value, such MFLINK requests may roll out while waiting for the pack to be mounted (provided the user specified the NA or WB parameter). Since there is no global resource executive in an LCN environment, a potential of a temporary deadlock exists in the latter instance. If this happens, the RHF applications involved will be timed-out and aborted.
DFPT	DI1	Equipment type. When accessing an auxiliary device with a permanent file command, the permanent file manager checks that the equipment type and pack name of the device match the equipment type (R parameter) and pack name on the command. If R is not specified, the system uses the equipment type specified by DFPT. If the default is used for another equipment type, the error message INCORRECT DEVICE REQUEST occurs.
FPXL	7777_8	Permanent file password or permission term limit in days, from 1 through 7777_8. This value establishes the upper limit on the expiration term that a user can specify on a permanent file request. Changes to this parameter should be supplied by a site security administrator.
FPXT	7777_8	Default permanent file password or permission term in days, from 1 through 7777_8. This value is assumed when the user does not specify an expiration term parameter on a permanent file request. FPXT must be less than or equal to FPXL. A value of 7777_8 indicates the password or permission cannot expire. Changes to this parameter should be supplied by a site security administrator.

Parameter	Default	Significance
MNHS	5	Minimum size hole, in sectors, that permanent file manager (PFM) creates in the indirect access file chain when using an existing hole. If, in the search for a hole in which to save an indirect access file, PFM finds that the use of an existing hole creates a new hole containing fewer sectors than MNHS, then PFM allocates space at the end of the indirect access chain. If a delink operation creates a hole smaller than MNHS, PFM delinks one less track to ensure minimum size for the hole. The purging of a file whose total length is less than MNHS results in the creation of a hole smaller than MNHS. If a value for MNHS is smaller than the average length of the indirect access files on the system, it results in holes that may be unusable. If the value is larger than the average file length, it results in holes which are not used for a period of time. For efficient use of holes, the value for MNHS should be close to the average length of the indirect access files on the system. The minimum value for MNHS is 3; the maximum value is 7777_8.
BRDE	BRAL	Backup requirement (BR) default specifications; can be set to the following symbolic values. **Value** — **Description** BRAL — Backup always required. BRMD — Media-dependent backup for systems with an alternative mass storage facility. BRNO — No backup required.
RSDE	RSNP	Preferred residence (PR) default specification; can be set to the following symbolic values. **Value** — **Description** RSMS — Mass storage facility residence preferred. RSNP — No preferred residence.
PMLM	62_{10}	Number of explicit permissions allowed per file, 1 through 4095. If PMLM is changed, PFM must be reassembled.
PGUI	300000_8	PFDUMP (refer to the NOS 2 Analysis Handbook) purge limit user index. When PFDUMP is used with the purge option, selected files under user indexes greater than PGUI are dumped, but not purged. If PGUI is changed, PFDUMP must be reassembled.

For individual users, each of four permanent file access limits is established through MODVAL (refer to the NOS 2 Administration Handbook) by specifying a range index from 0 through 7. Each range index corresponds to an upper limit specified by one of the following installation parameters. The last character of the installation parameter indicates the range index being defined. Table 5-8 summarizes the released values for each parameter. Setting a parameter to 0 indicates unlimited access.

Parameter	Significance
NFRNGn	Upper limit of range n for file count; must not exceed 77777_8.
CSRNGn	Upper limit of range n for cumulative size of indirect access files, specified in PRUs; must not exceed 777777_8.
FSRNGn	Upper limit of range n for size of individual indirect access files, specified in PRUs; must not exceed 77777_8.
DSRNGn	Upper limit of range n for size of individual direct access files, specified in PRUs; must not exceed 777777_8.

Table 5-8. Released Values of Permanent File Limit Ranges

Parameter	Values of n						
	1	2	3	4	5	6	7
NFRNGn	10	20	30	40	50	100	0
CSRNGn	1000	2000	5000	10000	50000	100000	0
FSRNGn	10	30	50	100	150	300	0
DSRNGn	1000	2000	5000	10000	50000	100000	0

All values are specified in octal; 0 indicates unlimited access.

COMSPRO Parameters

The following COMSPRO parameters contain a general description of the PROFILa file. Assemble CALLSYS to obtain a listing of COMSPRO.

Parameter	Default	Significance
PPFN	PROFILC	Micro definition specifying the PROFILE routine's database file name (refer to the NOS 2 Administration Handbook).
PPWD	SECURUS	Micro definition specifying the PROFILE routine's database file password.
PUSN	SYSTEMX	Micro definition specifying the catalog location of the PROFILE routine's database.

COMSREM Parameters

COMSREM contains the following parameters, which are used by the Interactive Facility (IAF) executive. Assemble CALLSYS to obtain a listing of COMSREM.

Parameter	Default	Significance
UTIS	10D	Value used in calculating the default CPU time limit in seconds for any particular terminal job's activity, if it is not specified with the SETTL command (refer to the NOS 2 Reference Set, Volume 3). For details of the algorithm used in calculating the default time limit, refer to the NOS 2 Administration Handbook.
VDSI VDTI	100B 100B	Default system resource unit (SRU) and time limit increment values for the S,nnnnn and T,nnnnn interactive commands.
VXLL	2500D	Maximum number of characters in a logical input line.
VXPH	2500D	Determines the physical line length that IAF accepts. IAF uses VXPH to calculate a buffer length.
TAPC	10B	Number of pots of typed ahead input to be stored in IAF's FL for each user.
VNCP	40B	Number of pots of output to be stored in IAF's FL for each user.
NMFL	60000B	Defines the size of NAM's field length as used by the algorithm in COMPCMX. This calculation determines the field length available for an interactive job. This value should be equal to the value of MAXNIP, which defines the maximum field length that NAM can attain. MAXNIP is an installation parameter for NAM.

COMSRSX Parameters

COMSRSX contains the following parameters, which are used by the resource executive. Assemble CALLSYS to obtain a listing of COMSRSX. The released values assume the default job scheduler cycle of 1 second.

Parameter	Default Value in Minutes	Significance
RPMS	4	Length of time that a job waiting for an auxiliary device is rolled out before retrying assignment.
RPOV	8	Length of time that a job that has had a request for an auxiliary device denied because of overcommitment deadlocks is rolled out before retrying assignment.
SUBM	10	If MAGNET is not active, length of time that a noninteractive service class job calling RESEX is rolled out before retrying assignment.
MTMS	2	When one of the following situations prevents immediate assignment of the tape, MTMS specifies the length of time that a job requesting a tape is rolled out before retrying the assignment. • The job requests a tape with a VSN that is not currently available. • The job requests a 9-track tape that is mounted without a write ring, the job requests the wrong tape density, and the tape hardware detects that the density of the tape is incompatible with the unit on which it is mounted (800-cpi tape on a 1600/6250-cpi drive or 6250-cpi tape on an 800/1600-cpi drive).
RFTL	10	Length of time that a job requesting a resource is rolled out before retrying the request when a track limit occurs on the resource demand or VSN files.
MTOV	8	Length of time that a job that has had a request for a magnetic tape denied because of overcommitment deadlocks is rolled out before retrying the assignment.

COMSSCD Parameters

COMSSCD contains the definitions of the service classes used by NOS. Assemble CALLSYS to obtain a listing of COMSSCD. By default, four installation service classes are defined: I0, I1, I2, and I3. To define additional installation service classes, add CLASS macro calls similar to those already present. You can also change the attributes of the defined installation service classes (those that are parameters on the CLASS macro). However, if you make any changes, reassemble all the decks that call COMSSCD.

COMSSFS Parameters

COMSSFS parameters are used by the MODVAL and PROFILE commands. Assemble CALLSYS to get a listing of COMSSFS.

Parameter	Default	Significance
FLLM	50000_8	Specifies the field length limit for the execution of the MODVAL and PROFILE commands. If the execution of a MODVAL or PROFILE command requires more than the specified field length, disk storage is used. Accessing disk storage is more time-consuming than accessing central memory and degrades performance.

COMSSRU Parameters

COMSSRU contains the parameters used in SRU calculations. Assemble CALLSYS to obtain a listing of COMSSRU. Refer to COMSSRU in the NOS 2 Analysis Handbook.

COMSSSJ Parameters

COMSSSJ contains the following parameters, which are used by special system jobs. Assemble CALLSYS to obtain a listing of COMSSSJ. The released values assume the default job scheduler cycle of 1 second.

Parameter	Default	Significance
ART	4 minutes	Length of time that a job is rolled out while waiting for a direct access file to become available before trying to access it again. This value specifies the default for the WB parameter on the ATTACH command.
FRT	15 seconds	Length of time that a special system job is rolled out when a fast attach file is busy.

COMS1DS Parameters

COMS1DS contains the following parameter used for DSD jobs. Assemble CALLSYS to obtain a listing of COMS1DS.

Parameter	Default	Significance
SYSCHG	SYSTEM	Micro definition that specifies the system charge number for jobs initiated by DSD.

COMTNAP Parameters

COMTNAP defines a table that maps valid network application names to the bit position of the access word that must be set to allow use of one or more network applications (refer to the description of MODVAL in the NOS 2 Administration Handbook). Assemble CALLTAB to obtain a listing of COMTNAP. This common deck does not contain any executable code. Each table entry has the following format.

59	17	11 0
application name (6-bit display code, left-justified, blank-filled)	reserved	application validation word bit position

Bits 17 through 12 of each entry are reserved for the program that uses COMTNAP. These bits are set to 0 when COMTNAP is assembled. The last word of the table must be 0.

Each application defined in COMTNAP must appear only once. However, any application validation bit can appear more than once; that is, a given application validation bit can be defined to permit use of more than one application, if the operations at a particular site make such a definition desirable. Bits 47 through 36 of the application validation word are reserved for customer application use; bits 35 through 15 are reserved for Control Data application use.

The released table defined by COMTNAP follows:

Word	Contents 59	17	11	0
TNAV TIAF	IAF		0	0
	RBF		0	1
	TAF		0	2
	MCS		0	3
	TVF		0	4
	CS		0	5
	PLATO/NAM Interface		0	6
	ITF		0	7
	PSU		0	1
	reserved bit 8		0	0
	reserved bit 9		0	0
	reserved bit 10		0	0
	reserved bit 11		0	0
TLAV	AP1		0	12
	AP2		0	13
	AP3		0	14
TNAV+ TNAVL	0			

The following table-related parameters are defined in COMTNAP. All symbols are unqualified.

Parameter	Default	Significance
TNAV	-	Table first word address. Program that uses COMTNAP defines the value.
TNAVL	17_8	Table length, excluding zero-word terminator.

COMUSIT Parameters

COMUSIT contains the following parameters, which are used by the ASMOVE utility to control selection of files for destaging to the mass storage facility (MSF) and for releasing their disk space. Compile CALLMSS to obtain a listing of COMUSIT. Compile ASMOVE to determine how the following weight and scale factors are used in the selection algorithm.

Parameter	Default	Significance
DEFDB	2	Weight factor for MSF-preferred residence.
DB$SCALE	1.0	Scale factor for MSF-preferred residence.
DEFDC	1	Weight factor for no preferred residence.
DC$SCALE	1.0	Scale factor for no preferred residence.
DEFDL	1	Weight factor for file length.
DL$SCALE	1.0	Scale factor for file length.
DEFDT	0	Weight factor for time since last modification.
DT$SCALE	1.0	Scale factor for time since last modification.
DEFDV	24	Weight factor for destage control value.
DV$SCALE	25.0	Scale factor for destage control value.
DEFMN	6	Weight factor for minimum file length.
MN$SCALE	25	Scale factor for minimum file length.
DEFMX	128	Weight factor for maximum file length.
MX$SCALE	250	Scale factor for maximum file length.

COMXIPR Parameters

COMXIPR contains the following parameters used by MAS executive (SSEXEC). Assemble CALLFAS to obtain a listing of COMXIPR.

Parameter	Default	Significance
NUMSLV	2	The number of slave mainframes for which master SSEXEC can service file staging requests.
NUMRB	9	The number of file staging request blocks available to a slave mainframe.
SLRP$INTV	5	The number of seconds SSEXEC waits to look for new staging requests from slave mainframes.
SLAV$INTV	60	The number of seconds SSEXEC waits with no signal from a slave mainframe before assuming that the slave executive has terminated.

PPCOM Parameters

PPCOM contains the following parameters, which are used by system peripheral processor packages for intercommunication. Assemble PPTEXT to obtain a listing of PPCOM.

Parameter	Default	Significance
UEBS	100_8	Number of words in a block of user ECS (1000_8, 2000_8, 4000_8).
ESMX	512D	Maximum number of EST entries, plus one. The value for ESMX can range from 10 through 512.
LCOM	12_8	Maximum length of the L-display input buffer in words. The value for LCOM can range from 1 through 12_8.
LDSY	350_8	Maximum length of the L-display output buffer in words. The value for LDSY can range from 100_8 through 1000_8.
MSMX	200D	Maximum number of mass storage devices. The value for MSMX can range from 1 to 200.

The assembly constant INSP$ is defined in NOSTEXT. If the INSP$ reference is deleted, 10 bytes in both the DSD display and the command overlays are unavailable for site code.

DSD Parameters

The following parameters, specified in ENTER macro calls (within the DSD syntax tables), cause the first 25 characters of the associated DSD command to be logged in the system dayfile and/or the error log. The commands are logged just as they are entered by the operator except that the characters

DS,

are placed before each command. The DSD listing contains an explanation of the ENTER macro. Assemble DSD to obtain a listing.

Parameter	Default	Significance
SDF	-	When specified in an ENTER macro call, the associated command is logged in the system dayfile.
ERL	-	When specified in an ENTER macro call, the associated command is logged in the error log. On the release tapes, the OFF, ON, channel control, and memory commands specify ERL on their ENTER macro calls.

NSS - NOS-SCOPE 2 STATION FACILITY 1

There are no special build instructions for NSS. For information on logical mainframe identifiers (LIDs), refer to Remote Host Facility (RHF), later in this section. For information about initiating SSF, refer to Subsystem Initiation, at the beginning of this section. This section describes these installation options:

- Spun-Off Task (SPOT) Memory Requirements
- NSS Validation Requirements
- Installation Parameters

SPUN-OFF TASK (SPOT) MEMORY REQUIREMENTS

The station control point MFSTAT handles all communication between the station and SCOPE 2. A SPOT is a job that MFSTAT initiates and places into the host mainframe input queue after MFSTAT detects a request for an I/O transfer or a staging operation. There are five unique SPOTs: permanent file SPOT, tape staging SPOT, spooling SPOT, dump SPOT, and deadstart SPOT. MFSTAT and all the SPOTs execute on the NOS mainframe.

The spooling SPOT (SOT76) is the only SPOT that transfers more than one file. SOT76 is initiated when communication is established between mainframes and does not terminate until the station is dropped. All other SPOTs are initiated as required to transfer one file, and each is terminated after completion.

Field length requirements fluctuate as file transfers are initiated and terminated. As file transfers are terminated, buffers are released. Definitions in the common deck COMTUNE control the lengths of the buffers used by the SPOTs. Each SPOT uses one of the COMTUNE SYMPL DEFs to determine the length of the I/O buffers. All SPOTs are written in the SYMPL language.

Nominal release definition octal values for the I/O buffers are described in the following list.

Name	Value	Description
LBUFDSTP	3004	Deadstart via tape
LBUFDSLM	7725	Deadstart link buffer
LBUFDP	4001	7000 dump
LBUFLM	1401	Link medium

Name	Value	Description
LBUFPF76	5051	Permanent file for 6000-7000
LBUFSP	2021	Spooling
LBUFTP	6007	Tape staging

Buffers required to transfer a file vary among SPOTs. All SPOTs transfer a file between a NOS disk and a link medium device. For some SPOTs, this requires two separate buffers as data is converted. Other SPOTs use only one buffer. When two buffers are needed, the length of the link medium buffer is defined by LBUFLM, and the buffer for the disk is defined by the appropriate symbol (for example, LBUFSP for the spooling SPOT).

Octal memory requirements for the various SPOTs are shown in table 5-9.

SPOTs, like user jobs, may be rolled or swapped out as memory is needed.

The relationship between buffer sizes and performance is bound by the same consideration as for any user job. The absolute minimum buffer size is a PRU + 2 words. Any large reduction in buffer sizes from the release values has some impact on performance.

Table 5-9. SPOT Octal Memory Requirements

SPOT Type	Code	Buffer Lengths Used to Transfer Files	Total Memory Used	Notes†
Spooling	10200	LBUFSP+LBUFLM	14600	Single file transfer
Permanent file	5100	LBUFPF76+LBUFLM	11600	
Tape staging††	6700	3*MBL+LBUFLM/2+3	11100	For MBL<(LBUFTP-LBUFLM/2-3)/3 [7690]
		LBUFTP	14700	For (LBUFTP-LBUFLM/2-3)/3 [7690] <MBL<LBUFTP-LBUFLM-1 [23090]
		MBL+LBUFLM+1	20500	For MBL>LBUFTP-LBUFLM-1 [23090]
Dump	4200	LBUFDP	10200	
Deadstart	6200	LBUFDSTP+LBUFDSLM	21200	Via tape

†Figures in brackets are release values for the equations.

††The buffer length required for tape staging depends on the value specified for the maximum block length (MBL) for tape staging operations. Use the equations in the Notes column to determine the appropriate values to use. The maximum value that can be specified for MBL is 50000.

NSS VALIDATION REQUIREMENTS

Add the SPOT user name entry (UN=SSPOT, PW=STATION) to the NOS validation file. This has been tested with the following validations, some of which may not be necessary:

MT=7	DF=77B
PR=7	SL=77B
SC=50B	AW=CLPF, CSPF, CCNR, CASF, CAND, CSRP, CSAP
MS=77B	CS
CM=20B	CC
TL=77B	CA
CP=77B	CA
DB=7	CS
LP=77B	CS
CC=77B	

If you do not have a validation file, SYSGEN(FULL) automatically creates the necessary validations. The released user name and password are SSPOT and STATION. To change the user name or password, create a USER file which applies a modification set to change line:

NSSA000.4601

Then rerun the NSS build procedure.

INSTALLATION PARAMETERS

The following installation parameters are all on common deck COMTUNE on the NOS station PL. Deck COMTUNE also lists the minimum, maximum, and default values for these parameters.

Table Size Parameters

NMF	The number of mainframes to which the station can be linked; the released value for NMF is 1, and the maximum value that can be specified is 2.
DATAENT	The maximum number of active data streams.
SNTS	The size of the SPOT name table (SNT). The size of the SNT should be the value of DATAENT plus the maximum number of local SPOTs (four) plus the spooling SPOT (one).
IDTMAX	The maximum size of the IDT table controls the number of logical IDs a mainframe can have and allows this value, minus two logical IDs. IDTMAX must match IDTL defined in PP program SSH; otherwise, SSH hangs the spooling SPOT until the station is dropped.
MAXSPOTS	A group of parameters that define the default maximum number of active SPOTs of each type that MFSTAT activates at one time.
SPOOLS	The maximum number of spooling streams.
DISNAMESIZE	The size of the display name table for the transparent display interface to the SCOPE 2 operating system.
SYNTAXSIZE	The size of the syntax extension table for the transparent display interface to the SCOPE 2 operating system.

Spooling and Recalling Time Parameters

MSEC(i)	Recall time in milliseconds.
SEC(i)	A method of approximating the number of busy station loops in i seconds.
ISEC(i)	A way of approximating the number of idle station loops in i seconds.
TME7000	The delay in seconds between sending time requests to the SCOPE 2 operating system.
STA7000	The delay in seconds used by MFSTAT between sending status requests to the SCOPE 2 operating system.
RCL7000	The recall time in milliseconds used by the busy overlay of MFSTAT when communicating with a linked SCOPE 2 operating system.
TIMEOUT	The length of time in seconds that MFSTAT waits before logging out a linked mainframe when communication is lost.
MSGCNT	The length of time in seconds that MFSTAT leaves informative messages on the B display.
BSYLIM	The length of time in seconds the busy overlay of MFSTAT delays after sensing an idle condition, before going into an idle state.
IDLRCLTM	The recall time in seconds used by MFSTAT when the idle overlay is executing.

NOTE

For better response time, lower both the RCL and STA values. To reduce CPU use, increase the RCL value. If the RCL and STA parameters are reduced too much, however, the reduction may cause STD (the link medium coupler driver) to be locked in.

LOOPLIM	The delay in seconds that MFSTAT waits before checking for a change in busy to idle status (controls the frequency with which the busy portion of the station checks its busy status).
DSDWAIT	The length of time in seconds that MFSTAT waits for a reply before it rejects a DSD request.
MAXINCOUNT	The frequency in seconds with which MFSTAT checks the input queues for files to spool when it is idle.
OVLMAX	The maximum time in seconds that MFSTAT retains the secondary overlay field length after a load of a secondary overlay.
IDLEMAX	The elapsed time in seconds that MFSTAT waits after all spooling activity has completed before swapping out the spooling SPOT.

SPLLIM	The number of seconds MFSTAT waits before going idle once spooling activity is completed.
IDLETIME	The number of seconds the spooling SPOT waits before trying to initiate spooling operations.
IDLETIME2	The amount of time in seconds the spooling SPOT waits before initiating new spooling operations if output spooling is taking place.
SPOOLRCL	The recall time in milliseconds used by the spooling SPOT when there is no spooling activity.

Buffer Size Parameters

DAYBUFSIZE	The size of the MFSTAT buffer for processing SPOT dayfiles on the active side.
RRBUF	The size of the MFSTAT active transmit buffer and the linked staged packet buffer.
LRGBUF	The size of the MFSTAT receive buffer for the active side of the station.
LICRBUF	The length of the MFSTAT buffer used by the IAF queue utility helper.
LBUFLM	The length of the link buffer used by SPOT jobs for NOS-SCOPE 2 I/O spooling.
LBUFSP	The length of the disk buffer used by the spooling SPOT for NOS-SCOPE 2 spooling of I/O files.
LBUFPF76	The length of the disk buffer used to read and write the disk for permanent file transfers to and from SCOPE 2.

The following installation parameter is on the deck HELLO7:

HELLO7 Multimainframe Flag

NOMFF	If NOMFF is set to 1 (the release value), HELLO7 supports a single mainframe. If multiple mainframe support is desired, NOMFF must be set to 0.

QU3 - QUERY UPDATE VERSION 3

The installation tool SYNGEN, and the various common decks that reside on the DDL 3 program library, must be available for the installation of Query Update 3. Following are the unique parameters for QU3.

IMF — The QU3 installation procedure provides you with the option of installing the Information Management Facility Version 1 (IMF 1) interface. To install Query Update 3 with the IMF 1 interface, specify the IMF keyword in the call that executes the QU3 procedure. IMF 1 must be installed prior to installing Query Update 3 with the IMF 1 interface. (The installation of IMF 1 is necessary only if you select the IMF interface option.)

SRT4 — To assemble code to interface with Sort/Merge 4 rather than Sort/Merge 5, specify the keyword SRT4 on the call to QU3. SRT4/SRT5 must be installed prior to installing Query Update 3 with the SRT4/SRT5 interface.

The QU3 interface to the Database Utilities, version 1, for CRM logging is provided by releasing the DBU relocatable binaries needed to load the QU/CRM logging capsule, CAPLOG, on the permanent file tapes. SYSGEN(SOURCE) loads this file to the installation user name INSTALL. The QU3 installation deck contains commands to access this file and to create a temporary library, DBULIB, to be used at load time. The deck also contains commands to add the command-callable portion of DBU (DFRCV) to the PRODUCT file, thus allowing inclusion of DFRCV in the deadstart tape. No special definitions are required to access this interface.

RBF5 - REMOTE BATCH FACILITY VERSION 1

This subsection describes the following installation options for RBF5:

- Unique Parameters
- RBF Command
- USER File Directives
- File Placement
- Verification

UNIQUE PARAMETERS

NOTRACE To disable log file creation for RBF5, specify the keyword NOTRACE on the call that executes the RBF5 procedure.

RBF COMMAND

RBF5 requires the following command.

RBF2PO(MC=mc)

Parameter	Description
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if mc is 0. This parameter is optional; no NETREL call is issued if MC is omitted. If NETREL is to be called, a file NRF1 must be assigned to the control point before the command is executed. File NRF1 must contain a valid job record for writing to the ZZZZZDN file. The released RBF job skeleton record creates NRF1 from the job input record.

USER FILE DIRECTIVES

To assemble various features into RBF, include directives of the form:

```
*DEFINE name
```

on file USER for the RBF5 installation job.

name	Significance When Defined
DEBUG	Code to aid in debugging and maintenance is generated.
IMS	Descriptive internal maintenance comments are included in the assembly and compilation listings.
TRACE	Symbolic table dumps of RBF are written to file SPITOUT when RBF fails.

The following parameters are defined in the common deck IP$COM. To make changes to these parameters, place appropriate Update directives on file USER for the RBF5 installation job.

Parameter	Default (decimal)	Significance
SEARCHTIME	15	Time interval in seconds between scans of the output queue for remote batch files. These times are increased by approximately 10 seconds when the load on RBF is light and when most of RBF's field length is rolled out to disk.
RESUMETIME	20	Time interval in seconds between receipt of the last interactive message and the automatic switching of the terminal to batch mode; should be larger than SEARCHTIME.
REFRESHTIME	30	Refresh period in seconds for the RBF console queue displays when RBF is specified on the DISPLAY command; should be larger than RESUMETIME.
STATIONS	16	Maximum number of consoles.
TOTDEV	32	Maximum number of batch devices.
MAXFL		Maximum field length to which RBF expands when obtaining buffers. If TRACE is defined, the default value is 100000; if TRACE is not defined, the default value is 50000.
ALOTIME	600	Time in seconds that a dial-in terminal is allowed to remain inactive before being timed out of RBF. A value of 0 specifies that terminals are not timed out of RBF. The maximum value allowed is 4095.

FILE PLACEMENT

Execution of the RBF5 installation procedure requires a prior build of NAM and the existence of the NAM startup master file under the build user index. The RBF5 installation procedure modifies the startup master file (NAMSTRT). Because other products may require the startup master file, refer to NAM5 File Placement in this section to determine when and where to move the file.

VERIFICATION

Use the following procedure to verify correct installation of RBF.

1. Prepare the following card deck. If necessary, supply the CHARGE command.

   ```
   job command.
   USER,username,password,familyname.
   NOTE.+1TOP OF PAGE+0ABCDEFGHIJKLMNOPQRSTUVWXYZ
   6/7/8/9
   ```

   A multipunch in column 1 (or for HASP terminals, /*EOI in columns 1 through 5).

2. Initiate NAM as described under NAM5 - Network Access Method. (It is assumed that RBF is defined as an application in COMTNAP.)

3. Log in from the console of any network-supported terminal with at least one card reader and one line printer, specifying RBF as the application and any user name permitted to use RBF (refer to the Remote Batch Facility Reference Manual). Follow all commands issued in this procedure by the message transmission key for the terminal class you are using (ETX or SEND for many batch terminal consoles.) RBF responds with a header line giving the version of RBF and the date and time of login, followed by READY on the following line.

4. Enter DIS. RBF displays the status of the batch devices associated with the terminal.

5. Place the card deck prepared in step 1 into the input hopper of the card reader and initiate reading (described in the Remote Batch Facility Reference Manual).

6. Enter the following command at the terminal console.

   GO,CRn

   n is the device number of card reader (from 1 through 7).

7. RBF reads the job in the card reader and responds with READY.

8. Enter DIS,PR at the terminal console. Because the system processes the job prepared in step 1 in little time, RBF should indicate that the job is in the print queue. If not, repeat this step until such an indication is received.

9. Ready the line printer, and enter the following command at the terminal console.

   GO,LPn

   n is the device number of line printer (from 1 through 7).

10. RBF prints the job's output on the line printer. The output consists of a banner page, a page with TOP OF PAGE printed on the first line of the page and the alphabet on the third line, and a page containing the job dayfile.

11. Enter BYE to exit RBF and log off of the system. The system issues a message indicating the time that has elapsed since connection to RBF.

    RBF ENDED yy/mm/dd. hh.mm.ss

    RBF CONNECT TIME hh.mm.ss

RHF - REMOTE HOST FACILITY VERSION 1

This subsection describes the following installation options for RHF:

- RHF Configuration File
- RHF Procedure File
- Hardware Configuration
- Installation Parameters
- Logical Identifier Definition and Use
- Loopback Capability
- Store and Forward Capability

RHF CONFIGURATION FILE

You must create an RHF configuration file using the Network Configuration Statements. You should define the RHF configuration including all NADs, applications, and physical identifiers (PIDs) to be used by or accessible to RHF. Refer to the NOS 2 Analysis Handbook for information about creating an RHF configuration file.

RHF PROCEDURE FILE

The entry point of the RHF subsystem is named RHF. Therefore, you can initiate RHF without a procedure file by entering the DSD entry RHF.

If you want to initiate RHF with a procedure file, create an RHF procedure file. If you do, that file must contain a RETURN,RHF command to return the local file RHF before the call to RHF. For example,

```
.PROC,RHFffff,...
RETURN,RHF.
local site commands
RHF.
```

RHFffff can be a procedure file stored in an indirect access permanent file under the system user index (377777_8).

Refer to the beginning of this section for more information about subsystem initiation.

HARDWARE CONFIGURATION

RHF and its applications require the same minimum hardware configuration as NOS plus a minimum of one 380-170 Network Access Device (NAD). NADs must not be physically connected on channel 0.

Switch settings on the NAD are critical. Many switch settings (such as access code, NAD address, TCI enable, and so on) must be correct to obtain any response from the NAD. The RESYNC and CONTENTION parameters, if not set properly, can cause occasional trunk errors. For example, if two NADs connected by one trunk have the same RESYNC parameter, a file transfer in one direction may fail with a broken connection. Set the CONTENTION/RESYNC parameter as follows: on any given trunk, the RESYNC parameter for each NAD should be unique and should be less than the value (2*contention number) + 2. Refer to the 380-170 NAD Hardware Reference Manual for further information.

INSTALLATION PARAMETERS

Parameter	Description
MAXFILEXFR	The maximum number of file transfers that the Facility Interface Program (FIP) allows for any one application. Values may range from 1 through 10. Default is 4. The current definition format is:

```
1       7            22
        DEF MAXFILEXFR  #4#;
```

The change deletion location is in common deck COMADEF. The following is an example of a parameter change.

```
DEF MAXFILEXFR  #5#;
```

Parameter	Description
FETBUFSIZE	The number of words assigned to buffer space for each file transfer. Values may be zero or greater, but FIP overrides low values. The current definition format is as follows:

```
1       7            22
        DEF FETBUFSIZE  #3200#;
```

FETBUFSIZE	Assigned (binary)	Assigned (coded)
0 to 532	532	992
532 to 992	532 to 992	992
992 and up	992 and up	992 and up

The default value for FETBUFSIZE is 3200, which corresponds to about 49 PRUs. Values larger than 6400 (98 PRUs) do not increase transfer rates appreciably, and make job swapping more likely because of the increased central memory required.

The change deletion location is in common deck COMADEF. The following is an example of a parameter change:

```
DEF FETBUFSIZE #2800#;
```

MLTF Initiation Procedure Parameters

RHF uses the system procedure MLTF (maintenance error logging transfer facility) to initiate periodic NAD error logging. (Routine NETLOG in the MLTF procedure does the actual recording of NAD errors in the binary maintenance log (BML).) The parameter used by the MLTF procedure controls the interval length between MLTF executions.

The default procedure definition follows:

```
.PROC,MLTF,DEL=10.
```

Parameter	Description
DEL=del	Time in minutes between MLTF executions (0 < del < 2048), assuming a job scheduler cycle of 1 second.

The change deletion location is in deck MLTFPROC.

NAD Controlware Initialization Parameters

A set of initialization parameters must be loaded into NAD memory along with the NAD controlware as documented in the 380-170 NAD Hardware Reference Manual. These parameters are assembled into LOADBC and are appended to all NAD controlware loads. The default values provide for a maximum of 25 remote NADs, 24 paths, use of all available NAD memory, and no NAD buffer tracing. To change any of the initialization parameters, you must modify the default values and reinstall LOADBC. LOADBC subroutine CNP attempts to maximize NAD memory use by allocating as much memory as possible to NAD buffers. This automatic allocation is defeated if the NAD memory size initialization parameter is set to nonzero. This parameter should not be changed without a thorough understanding of NAD controlware memory use.

LOGICAL IDENTIFIER DEFINITION AND USE

The logical identifier (LID) is the identifier used to refer to a mainframe. A user refers to an LID on an MFLINK command, an MFQUEUE command, a ROUTE command, a Job command, or an ITF prompt.

For a successful access to or from another mainframe, both the RHF configuration and the LID table in CMR must be set up correctly. The other (remote) mainframe likewise must have its configuration and LID tables set up properly to receive or generate a successful network access.

For either QTF, MFLINK, or ITF to initiate access to a remote mainframe, the LID must be defined and enabled in the LID table in CMR. This can be done either through the LIDCMid file entries or through the operator utility LIDOU (refer to the NOS 2 Analysis Handbook).

For either QTFS or PTFS to respond to a remote mainframe request for file transfer, the remote mainframe's PID must be defined and enabled in RHF's configuration table as a valid PID for that mainframe. Likewise, the LID specified on the remote mainframe must be defined and enabled under that mainframe's host PID in the CMR LID table.

LOOPBACK CAPABILITY

A special loopback capability is available on RHF for the use of both QTF and MFLINK. This capability is intended primarily for test purposes, but it may be used for other purposes such as files across families on the same mainframe. The loopback capability allows a file to be sent from the local mainframe out to a NAD and back to the local mainframe instead of to a remote mainframe. To use this capability, the LIDs used to specify loopback must be defined properly in the CMR LID table. Refer to the NOS 2 Analysis Handbook for information about the LIDCMid file and the LID table.

To allow loopback, a PID entry (and associated LID and PATH entries) must be defined to match the local mainframe (Mxx, where xx is the machine id) in the LID and RHF configuration files. The loopback path should be defined. The remote NAD specified on the PATH definition should have, as an address, the address of the local NAD.

The LID to be used for MFLINK and/or QTF must be defined in the CMR LID table as enabled and loopback, under the mainframe host PID.

STORE AND FORWARD CAPABILITY

A queued file can be transferred from one mainframe to a remote mainframe that is not directly linked to the other mainframe. You can do this by using the store and forward capability. Refer to the NOS 2 Analysis Handbook for information about the LIDMCid file and the LID table.

RHP - PTF/QTF FILE TRANSFER FACILITY

This subsection describes the following installation options for RHP:

- Unique Parameters
- Installation Parameters
- QTF Initialization Procedure Parameters

UNIQUE PARAMETERS

SUBSYS=subsys — To install the file transfer applications to interface with RHF or NAM or both subsystems, use the SUBSYS=subsys parameter. Here are the values you can specify for subsys:

RHF

NAM

BOTH

The default is SUBSYS=RHF.

TRACE — To enable AIP/FIP tracing, include the keyword TRACE on the procedure call.

DEBUG — To enable full debug code, include the keyword DEBUG on the procedure call. Unsupported code for debugging purposes when writing and testing RHF components is available by including the E and C parameters on all SYMPL compiler commands and PC=DEBUG on all COMPASS commands. This code is not normally compiled or assembled and is not intended for a production environment.

Table 5-10 shows the binaries that are built depending on which subsystem you have specified.

Table 5-10. Binaries Built

	SUBSYS=RHF	SUBSYS=NAM	SUBSYS=BOTH
PTF	MFLINK FTF0100 FTF0200	MFLINK FTF0300 FTF0400	MFLINK FTF0500 FTF0600 FTF0700
PTFS	PTFS PFS0100 PFS0200	PFTSN PFS0300 PFS0400	PFTS PFS0100 PFS0200 PTFSN PFS0300 PFS0400
QTF	QTFI QTF	QTFIN QTF	QTFI QTFIN QTF
QTFS	QTFS QFS0100 QFS0200	QTFSN QFS0300 QFS0400	QTFS QFS0100 QFS0200 QTFSN QFS0300 QFS0400
MFQUEUE	MFQUEUE	MFQUEUE	MFQUEUE

INSTALLATION PARAMETERS

For information about MFLINK requests and auxiliary pack options, refer to the APLO parameter under the NOS COMSPFM deck in this manual.

Parameter	Description
ACNMAXC	The maximum number of connections that queue file transfer facility (QTF) can have active at any one time. Values can range from 1 through 10; the default is 4. (Lower values reduce QTF's memory requirements, but may also reduce the number of queue files transferred simultaneously.) The current definition (acceptable to both COMPASS and SYMPL) is as follows:

```
1          11     18    24          36
#ACNMAXC   #DEF#  4     #ACNMAXC    #4#;
```

The change deletion location is in common deck COMCAPR. An example of a parameter change follows.

```
#ACNMAXC   #DEF#  2     #AACNMAXC     #2#;
```

TIMEOUT	The time in seconds (assuming a job scheduler cycle of 1 second) in which a response must be received by an application from the remote NAD/application before the connection is broken. Values may range from 1 through 1800 seconds. Default is 600. The current definition format (acceptable to both COMPASS and SYMPL) is as follows:

```
1          11     18    24         34
#TIMEOUT   #DEF#  600D  #TIMEOUT   #600#;
```

The change deletion location is in common deck COMCAPR. The following is an example of a parameter change.

```
#TIMEOUT   #DEF#  400D  #TIMEOUT   #400#;
```

MAXRTRY	The number of retries that an application attempts to successfully complete a file transfer using RHFAM. Values may range from 1 through 50. Default is 10. The current definition format (acceptable to both COMPASS and SYMPL) is as follows:

```
1          11     18    24         34
#MAXRTRY   #DEF#  03D   #MAXRTRY   #03#;
```

The change deletion location is in common deck COMCAPR. The following is an example of a parameter change.

```
#MAXRTRY   #DEF#  20    #MAXRTRY   #20#;
```

QTF INITIATION PROCEDURE PARAMETERS

RHF uses the system procedure QTF to initiate the RHF application QTF. The parameter used by the procedure controls the interval length between QTF executions.

The default procedure definition follows:

```
.PROC,QTF,DEL=10.
```

Parameter	Description
DEL=del	Time in minutes between job executions ($0 < del < 2048$), assuming a job scheduler cycle of 1 second.

The change deletion location is in deck QTFPROC.

SORT - SORT/MERGE VERSION 4

HARDWARE REQUIREMENTS

If you use the tape sort option, polyphase requires three additional magnetic tape units and balanced requires four.

USER FILE DIRECTIVES

Whether Sort/Merge 4 uses the CMU hardware depends upon the IP.CMU parameter (refer to TEXT and TEXTIO, later in this section) from IPTEXT. To override this parameter, make the following changes on file USER.

To install Sort/Merge 4 without CMU code (default):

```
*I,FEAT64.42

BDP.INST     EQU    BDP.NO
```

To install Sort/Merge 4 with CMU code:

```
*I,FEAT64.42

BDP.INST     EQU    BDP.YES
```

TAF - TRANSACTION FACILITY VERSION 1

For an overview of the TAF installation process, refer to the installation overview appendix in the TAF Reference Manual.

This subsection describes these installation options for TAF:

- Unique Parameters
- Task Library
- TAF Procedure File
- TAF Validation Requirements
- Installation Parameters

UNIQUE PARAMETERS

DEBUG — You can install TAF with or without the trace feature. To get the trace feature, specify DEBUG on the call to the installation procedure TAF.

TASKLB — To create a task library (refer to TASK LIBRARY below), specify the keyword TASKLB on the call to TAF installation procedure.

TASK LIBRARY

Before running the TAF installation procedure, create a task library permanent file containing the following required tasks:

Task	Description
ITASK	Initial task.
BTASK	Task that recovers transactions initiated by BTRAN.
CRMTASK	Task that formats TAF/CRM Data Manager file status displays.
CTASK	Task to recover transactions using the TAF/CRM Data Manager.
RTASK	Task to recover terminals. RTASK may be on the task library permanent file or on database libraries.
RCTASK	Task that recovers CDCS transactions.
KDIS	TAF K display driver.
LOGT	Task to log out transaction terminal from TAF.
MSABT	Diagnostic generator for abnormally terminating tasks.
OFFTASK	Inactive task controller.
STASK	Send message then cease.
SYSMSG	Message task for system origin messages.
XTASK	Execute named task.

If TAF is used in a multimainframe complex, the system does not allow concurrent access to the same database. A copy of TAF in each computer must have its own user name/user index or default family.

TAF PROCEDURE FILE

Refer to Subsystem Initiation at the beginning of this section for information about the TAF procedure file and about subsystem initiation.

The user name required by TAF (KB100DC) is created automatically by SYSGEN(FULL) if no validation file exists.

TAF VALIDATION REQUIREMENTS

If you have no validation file, SYSGEN(FULL) creates the necessary user names and passwords for running TAF--user name KB100DC and password TAFPASS under user index 16 octal. The user name is used by SYSGEN to install the released TASKLIB file and for TAF operations. To change the password, supply a USER directive in the TCF file. To change the user name or user index, reassemble TAF changing the USNM and TRUI micros. These micros are set in deck COMKIPR. If you want to use SYSGEN to load files to the user names, you must temporarily set the password to the default password or modify SYSGEN.

INSTALLATION PARAMETERS

Unless otherwise specified, the following parameters are defined in deck COMKIPR. These parameters specify the charge and project numbers and user index for TAF. They also specify the user name under which TAF runs.

Parameter	Default	Significance
CGNM	A null micro	Micro whose string specifies the charge number for TAF; used when a dump is performed. If CGNM is null, no CHARGE command is issued, and the user name specified by USNM must not require a CHARGE command.
PJNM	A null micro	Micro whose string specifies the project number for TAF.
TRUI	16B	User index for TAF.
USNM	KB100DC	Micro whose string specifies the user name under which TAF runs.

The following parameters specify the default initialization K display options.

Parameter	Default	Significance
ECSFL	0	Extended memory field length/1000_8. ECSFL cannot be less than 0 nor greater than 400_8.
NCMB	40	Actual number of communication blocks allowed in the subsystem. Communication blocks hold incoming terminal input. This parameter can be changed by the initialization command K.CMB, but it cannot be less than 19 nor greater than 40.
NSCP	31	Maximum number of subcontrol points. It cannot be less than 2 nor greater than 31.
SCMFL	376600B	Maximum field length. SCMFL cannot be less than 40000_8 nor greater than 376600_8, and must be set to a value that can be attained.
TLFM	TASKLIB	Micro whose string specifies the system task library file name.

The following parameters, defined in deck TAF, specify the default DSDUMP parameters. The user can override the parameters specified on CMDUMP requests with a task.

Parameter	Default	Significance
DEXP	1	Exchange package dump flag: Value — Description 0 — Exchange package is not dumped. 1 — Exchange package is dumped.
DFWA	0	First word address in octal for task dump.
DLWA	100000B	Last word address in octal for task dump.
DORC	BCOT	Origin code.
DORT	0	Output disposition (corresponds to OQ parameter on DSDUMP/CMDUMP requests): Value — Description 0 — Local batch output queue. 1 — Remote batch output queue. 2 — Direct access permanent file. Refer to the TAF Reference Manual for further information.
DSQID	0	Batch identification (ID) code for output of jobs entered in the input queue by the task SUBMT request. The system assigns this ID to the output from jobs containing a SETJOB,DC=DF command. DSQID ranges from 0 through 67_8.

The following parameters specify default time dependencies. Although these values are expressed in milliseconds, they are accurate to only 1 second.

Parameter	Default	Significance
CORTL	1*1000	How often TAF checks to see if memory can be released to the system.
ITRTL	1500	Time to wait for input before rollout of transaction executive field length. ITRTL is defined in deck TAF.
RRTTL	1*1000	Time allowed to elapse before evicting a reusable task.
TACTL	2*60*1000	Time allowed to elapse between TAF receiving any input and TAF generating a call to ITASK. TACTL is defined in deck TAF.
TROTL	10*60*1000	Duration of rollout. TROTL is defined in deck TAF.
DMMTL	4	Time allowed to elapse between calls to the data manager(s).
TSKTL	120	Task time slice in milliseconds.

The following parameters, defined in deck TAF, specify default task rollout parameters.

Parameter	Default	Significance
DWITL	8*60	Time in seconds that a task is allowed to wait for terminal input before aborting. The user can override this parameter with the WAITINP request.
NESTL	16	Nest limit for CALLRTN (must be less than 64).
RTDNL	2*1000	Number of milliseconds a task is allowed to remain in memory waiting for a CALLRTN to complete.

The following parameters specify other default TAF installation parameters.

Parameter	Default	Significance
DTSTL	16	Number of time slices for a task. The user can override DTSTL for an individual task with the ITL request. DTSTL is defined in deck TAF.
IPTAR	1	Automatic recovery flag:

Value	Description
0	Automatic recovery is disabled.
1	Automatic recovery is enabled.

If recovery is disabled, the following requests are not honored in recovery mode.

Request	Comments
CALLTRN	Transactions can be scheduled, but input is not logged to the communication recovery file (CRF).
RERUN	
RGET	
RPUT	
RSECURE	
SECURE	
TINVOKE	
TSTAT	Except for the keywords USER and NEXT.
WSTAT	Except for the keywords STEP (=8 or =9) and USER.

Parameter	Default	Significance
IPTST	500	Number of terminals that can access TAF. IPTST must be greater than 0 and less than 4095.
RECDF	0	Default user recovery flag:

Value	Description
0	User recovery is enabled.
1	User recovery is disabled.

Parameter	Default	Significance
DTYM	DI	Micro whose string specifies the device type for journal files.

Parameter	Default	Significance
IFL=	200000B	Initialization field length; defined in deck TAF. This value must be large enough to load the Application Interface Program (AIP) required for NAM interface, and the desired data managers and various tables required by TAF during initialization. If the message MEMORY OVERFLOW DURING INITIALIZATION is issued, either increase IFL= or decrease the databases, the number of data manager buffers, or the number of communication blocks.
MAXJL	2500	Maximum word count on one journal request to any journal file, including header words; defined in deck TAF.
MAXRA	500B	Task limit for RA+1 requests; defined in deck TAF.
MAXTO	6*MAXWS	Maximum number of words task can send to the communication subsystem. Reaching or exceeding this value causes the task to abort.
MAXWS	409+1	Number of words SEND can transmit plus 1. Exceeding this value causes the task to abort.
TLDL	TLDLE*10	Amount of space to reserve for added tasks in the TAF-resident copy of the directory of each task library attached by TAF. This space can be used when TAF is informed of a task library change through the LIBTASK TT option. The value of the symbol should be a multiple of the size of a task library directory entry (TLDLE, currently 3). The default value allows space for 10 (TLDL/TLDLE) additional tasks. If more than TLDL/TLDLE tasks are added by the TT option, only the first TLDL/TLDLE tasks can be executed. The next time TAF is reinitialized, however, all the tasks added via the TT option are available to be executed. TLDL is defined in deck COMKTLD.

The following parameters are used with the TAF/CRM Data Manager.

Parameter	Default	Significance
AOBFL	31	Output queue length.
AIBFL	31	Input queue length.
CMAXDB	31	Number of CRM databases.
CMDM	31	Maximum number of transactions concurrently issuing TAF/CRM Data Manager requests and the number of segments in each before-image recovery file belonging to TAF/CRM Data Manager. If you change this parameter, database recovery is not possible using existing before-image recovery files: you must recreate the before-image recovery files.
CMMBFL	50000B	Base field length in words for common memory manager (CMM) buffer management.
CMMEFL	0	Number of words for CMM to expand buffer management.
CMMTFL	30000B	The upperbound on CRM's target field length. This area within the CMM buffer is used by CRM data and index blocks (for more information, refer to appendix G of the CRM Advanced Access Methods Version 2 Reference Manual).
CMMCAP	35000B	Maximum space within the CMM buffer that CRM uses for loading capsules, and for internal tables plus a padding value. This value is used to determine whether the value of CMMBFL is set properly in relationship to CMMTFL and the needs of CMR based upon the file organizations to be used by TAF/CRM.
CRMUPM	15	Number of updates allowed. Also defines the number of records in each segment of the before-image recovery files.
BMAX	8	Number of before-image recovery files. The maximum value for BMAX is 63.
RMDM	1	Number of mainframes running TAF/CRM Data Manager.

The following parameters are used with TOTAL Data Manager and are defined in deck TAF. For information on the installation of TOTAL, refer to the NOS 2 Applications Installation Handbook.

Parameter	Default	Significance
TIMDM	10	Maximum number of transactions concurrently issuing TOTAL Data Manager requests.
TMAXDB	31	Maximum number of TOTAL databases that can be initialized.
TMAXFIL	100	Maximum number of files per database.

The following parameters are defined in deck COMKNWC.

Parameter	Default	Significance
MLIM	100	Maximum number of words in one SEND request before a task is rolled out pending completion of terminal output.
NCTL	250	Maximum number of terminals in network communication table (NCT). To reduce core storage requirements, NCTL may be less than the total number of terminals in the network file (each entry requires 3 CM words). NCTL should be greater than or equal to the maximum number of terminals logged in at one time. If NCTL is exceeded, a terminal is rejected upon login. If the number of terminals defined in the NCTFi file is less than NCTL, the number of terminals in NCTFi replaces the value specified by NCTL. NCTL is defined in COMKIPR.

NOTE

The installation parameter MAXRA must be equal to or greater than the value for NCTL for successful initialization of TAF. This is due to the processing that CTASK must complete for every user during initialization.

The following parameters specify the default communication block parameters.

Parameter	Default	Significance
CBDL	57	Length of the data input area in the communication blocks. This parameter is in deck COMKCBD.
CBUL	9	Length of user area in the communication blocks. This parameter is in deck COMKCBD.
NCBC	4	Maximum number of communication blocks reserved for large transaction input.
NLIN	4	Maximum number of users allowed to perform large transaction input simultaneously. TAF reserves n - NLIN x NCBC - RSCMB communication blocks for smaller transaction input. n is the number of communication blocks with which TAF is initialized. NLIN should not be less than 4.
RSCMB	2	Maximum reserved communication blocks for nonterminal use. This number is included in the NCMB parameter.

The following parameters, defined in deck COMKTLD, specify the default task library parameters.

Parameter	Default	Significance
TLDMT	600	Number of tasks per library. The maximum value for TLDMT is 1365.
TLDMN	10	Number of tasks that may be added online to TAF's copy of any particular task library directory by the LIBTASK TT option.
TRDMN	10	Number of transactions that may be added online to TAF's copy of any particular transaction library directory by the LIBTASK TT option.
TRDMT	300	Number of named transactions per library.

The following parameters, defined in deck DMREC (except where otherwise indicated), specify the default batch recovery parameters.

Parameter	Default	Significance
AAICL	200	Number of ignore entries.
CRMARB	15	Number of after-image records that are buffered in the CM buffer for the file before they are flushed to disk. Also, the block length for after-image recovery files (ARFs). If you change this parameter, you must dump and recreate all ARFs. This parameter is in deck COMKIPR.
CRMARFN	35000	Length in physical record units (PRUs) of after-image recovery files. When preallocated by TAF or DMREC, the length specified by CRMARFN is assigned to the files excluding the header. This parameter is in deck COMKIPR.
DTTP	1	Tape drive type definition for dumping database and after-image recovery files; defined in deck COMKIPR. **Value** — **Description** 0 — Seven-track tapes. 1 — Nine-track tapes.
EXPCT	10	Default value of the percentage parameter for the EXPAND directive of deck DMREC.
FTABL	5000	Length of intermediate ignore table used during DMREC recovery.
NCOPY	2	Number of backup dumps to keep.
NDUMP	100_8	Number of dumps or directives. NDUMP must be less than 500_8.
NUMARF	1	Number of duplicate ARF copies.
TDEN	0	Tape density for dumps; any of the following. **Value** — **Description** 0 — System default density 1 — 556 cpi 2 — 200 cpi 3 — 800 cpi 4 — 1600 cpi 5 — 6250 cpi This parameter is in deck COMKIPR.

Parameter	Default	Significance
TDTR	40_8+10_8*DTTP+TDEN	Tape format definition.
TTIGL	5000	Length of table that contains the transaction entries to be ignored duuring DMREC recovery.
TLOGL	100	Number of files in database.
TVSNL	40	Number of VSNs allowed.
WBUFL	4001B	Length in words of buffer used to contain data read from a block on the after-image recovery file. Size depends on the installation parameters CRMARB and CMDM, and on the maximum record length specified for the database files in the xxJ file (refer to the TAF Reference Manual).

TDU - TERMINAL DEFINITION UTILITY

To build the TDU, you must use the SES CYBIL compiler.

TEXT AND TEXTIO - PRODUCT TEXTS AND PRODUCT TEXTS I/O

General installation parameters related to the common products are defined within the common deck IPARAMS, included in Product Texts.

The default values of the IPARAMS configuration parameters are defined with the CEQU or CMICRO macros so that you can insert all modifications at one place. The CEQU and CMICRO macros define symbols conditionally; that is, they are effective only if the variables have not been previously defined. Therefore, any modifications you make must precede them. Insert all changes to IPARAMS at IPARAMS.15.

Modifications to be applied to products TEXT and TEXTIO should be applied only in the procedure TEXT.

To obtain a listing of all installation parameters in IPARAMS, run a job similar to the following:

```
job command.
USER,INSTALL,INSTALL.
BEGIN,PRDIN,INSTALL,PRDNAME=TEXT,DISK=0.
UPDATE,Q.
COMPASS,A,I,S=0.
--eor--
*COMPILE IPTEXT
--eoi--
```

UNIQUE PARAMETERS

CSET Defines the character set to be used throughout the system. The character set selected determines the collating sequence to be used; that is, the order in which records are retrieved from a database and the results of comparisons of characters on a basis of greater than or less than. Refer to the CYBER Record Manager Basic Access Methods Reference Manual for a description of the collating sequences. The default value is C64. Here are the allowable values:

C64	Selects the CDC graphic 64.
A64	Selects the ASCII graphic 64.
C63	Selects the CDC graphic 63.
A63	Selects the ASCII graphic 63.

This parameter sets the IP.CSET symbol; these products reference IP.CSET.

AAM2	FCL 5
ALGOL-60 5	FORTRAN 5
APL2	Sort/Merge 4
BASIC3	Query Update 3
COBOL5	Update 1
COMPASS 3	8-Bit Subroutines 1
FCL 1	
FCL 2	

ECS This parameter specifies whether extended memory is available for use by Loader. Allowable values are YES and NO. The default is YES since no negative impacts result if extended memory is not available. If you specify YES, extended memory is available for use by Loader when loading user programs; if you specify NO, user programs loaded by Loader cannot use extended memory. This parameter sets the IP.MECS symbol.

MFT This parameter selects the mainframe model type. All values of the model micro are supported. The default is for the CYBER 74. Here are the allowable values:

71, 72, 73, 74, 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 810, 815, 825, 830, 835, 845, 855, 865, or 875. This parameter sets the MODEL, HF.LIST, and IP.CMU symbols; most common products reference the MODEL micro. If you have an 840, 850, or 860, set MFT to 855.

NOTE

Do not add user code to set the symbols IP.CSET, IP.MECS, IP.CMU, MODEL and HF.LIST since they are set by the above parameters. To specify your own values for HF.LIST, IP.CMU, and the MODEL micro, specify MFT=0 and apply user code to set these symbols.

The following list constitutes the extent of installation-changeable symbols in IPARAMS.

Parameter	Default	Significance
IP.CMU		If a value other than zero is specified, the compare/move unit hardware is present; if you specify zero, the compare/move unit hardware is not present. A value for IP.CMU is set by the unique parameter MFT on the call for the TEXT installation procedure. The following products reference IP.CMU: COBOL5 Sort/Merge 4
IP.CSET		Defines the character set to be used throughout the system. This parameter is controlled by the unique parameter CSET on the call to the TEXT installation procedure. The character set selected determines the collating sequence to be used. To select the ASCII graphic 64-character set, specify a value of IP.C64.2 for the IP.CSET parameter. To select the CDC graphic 63-character set, specify the following two parameter definitions: IP.C63 EQU IP.C64.1 IP.CSET EQU IP.C63 To select the ASCII graphic 63-character set, specify two parameter definitions: IP.C63 EQU IP.C64.2 IP.CSET EQU IP.C63 The following products reference IP.CSET: AAM2, ALGOL-60 5, APL2, BASIC3, COBOL5, COMPASS 3, FCL 1, FCL 2, FCL 5, FORTRAN 5, Sort/Merge 4, Query Update 3, Update 1, 8-Bit Subroutines 1

Parameter	Default	Significance
MODEL		This parameter, which selects the mainframe model type, is controlled by the unique parameter MFT on the call to the TEXT installation procedure. All values of the model micro are supported. The default is for the CYBER 74. Here are the allowable values: 71, 72, 73, 74, 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 810, 815, 825, 830, 835, 845, 855, 865, or 875. This parameter sets the MODEL symbol; most common products reference the MODEL micro.
IP.MECS	1	If you specify a nonzero value, extended memory is available for use by LOADER when loading user programs. If you specify zero, user programs loaded by LOADER cannot use extended memory. This parameter is controlled by the unique parameter ECS on the call to the TEXT installation procedure.
OS.ID	NOS 2.4.1	System identification micro for displaying the operating system name and version number in generated program binaries. Most common products reference the OS.ID micro.

Parameter	Default	Significance
HF.LIST		Micro whose value specifies the presence of certain hardware features in the configuration on which the products are being used. You should supply HF.LIST in addition to the MODEL micro, since use of various hardware features by the products is conditional on HF.LIST. However, if you do not define HF.LIST, the system selects a default value that is based on the MODEL micro and assumes no optional hardware. The default HF.LIST is a temporary capability that will be removed in a future release. You can define the following entries in HF.LIST.

Entry	Description
C	Compare/move unit (CMU) hardware is present.
L	For model 176, LCME is present. For models 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, and 875, UEM is present only if defined during deadstart. Both LCME and UEM are kinds of memory for which direct access instructions (014 and 015) are defined.
Sn	Stack size; n specifies the size of the longest possible instruction stack program loop in words. If the mainframe being described has no stack, omit this entry. n can be either of the following.

7	74 and 6600
10	175, 176, 740, 750, 760, 865, and 875

Parameter	Default	Significance	
		Entry	**Description**
		Px	Type of central processor; x can be one of the following values. S — 6200, 6400, 6500, 71, 72, 73, 171, 172, 173, 174, 720, 730, 810, 815, 825, 830, 835, 840, 845, 850, 855, and 860; serial type CPU, etc. 74 — 6600, 6700, and 74 175 — 175, 740, 750, and 760 176 — 176 740 — 740 750 — 750 760 — 760 865 — 865 875 — 875 The processor type defaults to PS if HF.LIST is defined but the processor type is omitted.
PSD		The central processor's exchange package contains a PSD register; model 176 only.	
CRW		Central memory read/write operations are performed for 660/670 instructions; models 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, and 875.	

Here are the default values selected for the HF.LIST parameter as set by the unique parameter MFT on the call to the TEXT installation procedure:

MODEL Micro Value	HF.LIST Default String
71	PS
72	C,PS
73	C,PS
74	P74,S7
171	PS
172	C,PS
173	C,PS
174	C,PS
175	P175,S10
176	P176,S10,L
720	C,PS
730	C,PS
740	P740,S10
750	P750,S10
760	P76,S10
810	C,PS,CRW,L
825	C,PS,CRW,L
830	C,PS,CRW,L
835	PS,CRW,L
855	PS,CRW,L
865	P865,S10,CRW,L
875	P875,S10,CRW,L
Any other	PS

Duplicate parameter entries (such as two Px entries) are not allowed.

When defining HF.LIST for products intended to be run on more than one mainframe, you can use the central processor type PS, P74, or P175 and include stack size (even if some of the mainframes do not have a stack). You must not include C and L unless those features exist on all of the mainframes in the configuration. The resulting products do not necessarily perform optimally on any one of the mainframes, but they perform better on a parallel processor (such as a 175) if that processor type is set in HF.LIST.

UPDATE - COMMON MEMORY MANAGER VERSION 1, CYBER COMMON UTILITIES, AND UPDATE VERSION 1

The following Update features are available through assembly options. You can modify them by deleting the appropriate entry in the range UPDATE.703 through UPDATE.711. An attempt to use these features when the option is not assembled causes Update to issue error messages. For example, when PMODKEY is not set, the PULLMOD statement is not recognized as a legal directive.

Parameter	Default	Significance
DECLKEY	Enabled	Enables DECLARE directive.
CHAR64	Enabled	Declares 64-character set Update program library output.
PMODKEY	Enabled	Enables PULLMOD statement and G option.
AUDITKEY	Enabled	Allows audit functions.
EDITKEY	Enabled	Allows merge and edit functions.
OLDPLKEY	Enabled	Enables Update to read both old-style and new-style old program libraries.
EXTOVLP	Enabled	Enables detection of four types of overlap involving two or more cards in a correction set.
DYNAMFL	Enabled	Declares dynamic table expansion. When this option is assembled, Update automatically expands tables as required and dynamically requests NOS to change the user field length to accommodate the additional table area. At the end of the run, the field length is reduced to that requested by the user.

Common Memory Manager (CMM) uses symbol definitions from common deck CMMCOM. The symbols defined in IPTEXT that specify the operating system are also used. You can change the following CMMCOM installation parameters for CMM.

Parameter	Default	Significance
DEFVER	0	Defines which of two versions of CMM is to be used by default. Value / Parameter 0 — A version without error checking (FAST) is used. 1 — An error checking version (SAFE) is used.
FLF	2000B	When variable block code is not present (only fixed blocks exist), this value is used as a default by the field length reduction algorithm. The amount of free space above the highest fixed block is reduced to FLF central memory words.
FLINC	2000B	When field length is increased by CMM, this value is used as a default increase above the minimum amount needed.

XEDIT - XEDIT VERSION 3

Store the file XEDITH under user name LIBRARY. Use the SYSGEN(MOVE) utility (refer to appendix B) to move XEDITH. You must run SYSGEN(MOVE) at the system console.

GLOSSARY A

Account Dayfile

The account dayfile provides a history of system usage. It also provides information necessary for accurate billing and system usage and analysis.

APRDECK

The APRDECK (Auxiliary Mass Storage Parameter Deck) is a text record on the deadstart file. The entries in this deck specify flawed disk areas. A flawed area is one that cannot be read from or written to by the system.

ASCII

American National Standard Code for Information Interchange. The standard character set and code used for information interchange between computer systems.

Auxiliary Device

Mass storage device that is not part of a permanent file family. Auxiliary devices can contain direct or indirect access permanent files.

CDCS

Refer to CYBER Database Control System.

Central Memory (CM)

The main storage device whose storage cells (words) can be addressed by a computer program and from which instructions and data can be loaded directly into registers.

Central Processor Unit (CPU)

The high-speed arithmetic unit that performs the addition, subtraction, multiplication, division, incrementing, logical operations, and branching instructions needed to execute programs.

Channel Number

The number of the data channel on which a peripheral device controller can be accessed.

Character

Unless otherwise specified, references to characters in this handbook are to 7-bit ASCII code characters.

Checkpoint

The process of writing to a magnetic tape or mass storage file a copy of your job's central memory, the system information used for job control, and the names and contents of all assigned files that are identified in a CHECKPT request.

CMRDECK

The CMRDECK (Central Memory Resident Deck) is a text file on the deadstart file. The entries in this deck describe the central memory table sizes to be used by the system; this deck also specifies which EQPDECK, IPRDECK, APRDECK, and LIBDECK will be used by the system.

Coldstart

Procedure used to deadstart if the tape or disk controller has not yet been loaded with controlware or the controlware is not running.

Command

A sequence of words and characters that call a system routine to perform a job step. A command is sometimes called a control statement.

Communication Line

A complete communication circuit between a terminal and its network processing unit.

Communication Network

The portion of the total network comprising the linked network processing units. The communication network excludes the host computer and terminals and is approximately equivalent to the set of all network elements configured as part of the total network.

Communications Supervisor (CS)

A portion of the network software, written as an application program, that coordinates the network-oriented activities of the host computer and of the lines and terminals logically linked to it.

Control Point Number

The number of the control point to which a job is assigned while the job resides in central memory. The actual number of control points is an installation parameter. Before the job can execute, each central processor program must be assigned a control point.

Control Statement

Refer to Command.

Controller

Hardware device that connects channels to peripheral devices. For example, a tape controller might connect up to eight tape units to one channel.

CPU

Refer to Central Processor Unit.

CS

Refer to Communications Supervisor.

CYBER Database Control System (CDCS)

The DMS-170 controlling module that provides the interface between an application and a data base.

Data Channel

One of the 9 to 24 channels (12-bit) by which information passes between the peripheral processors and peripheral devices. Refer to Channel Number.

Dayfile

A chronological file created during job execution which forms a permanent accounting and job history record. Dayfile messages are generated by operator action or when commands are processed. A copy of the dayfile is printed with the output for batch jobs. You must explicitly request it in an interactive job.

Deadstart

The process of initializing the system by loading the operating system library programs and any of the product set from magnetic tape or disk. Deadstart recovery is reinitialization after system failure.

Deadstart Sequencing

The execution of a selected set of commands before normal system job scheduling is enabled.

Default Value

A fixed value supplied by the system for a missing parameter.

Detached Job

An interactive service class job removed from control of the interactive subsystem. It may or may not continue to execute, depending on the presence of commands in the command buffer or an active job step. Control is regained by recovering the EJT entry for the job.

Direct Access File

A NOS permanent mass storage file that can be attached to your job. All changes to this file are made on the file itself rather than a temporary copy of the file (compare with Indirect Access File).

DIS (Job Display)

A system peripheral processor program similar to system display (DSD) program. DIS provides communication between a job in central memory and the operator at the console, and permits the operator to control execution of the program through the console keyboard.

Disk

A unit composed of one or more flat, circular plates with magnetic material on both sides that is used to store large amounts of data or programs.

Disk Pack

A group of disks with magnetically encoded information.

Display Code

A 6-bit character code set used to represent alphanumeric and special characters.

Displays

Two console screens or a split screen used to display system and job information, operator messages, and contents of central memory. Through the console keyboard, the operator can control the operation of the system. The displays are identified by alphabetic characters; some used frequently are the job status (B), system files (H), and dayfile messages (A).

Disposition Code (DC)

A 2-character mnemonic indicating the destination queue and format for processing a file named on a ROUTE function.

DSD (Dynamic System Display)

The operating system program that provides communication between the operator and the system by accepting control information typed on the console keyboard and by displaying information pertinent to all jobs known to the system. DSD is permanently assigned to peripheral processor 1.

ECS

Extended Core Storage. Refer to Extended Memory.

EQPDECK

The EQPDECK (Equipment Deck) is a text file on the deadstart file. The entries in this deck describe the hardware connected to your computer. These entries are used by the system to build the equipment status table (EST).

Equipment Number

A number from 0 to 7 which identifies the setting on a peripheral device controller.

Equipment Status Table (EST)

A central memory resident table listing all defined equipment, parameters affecting their operation, and the status of the equipment.

EST

Refer to Equipment Status Table.

EST Ordinal

The number designating the position of an entry within the equipment status table (EST) established at each installation. Devices are identified in operator commands by EST ordinals.

Extended Core Storage (ECS)

Optional additional memory. ECS contains 60-bit words; it has a large amount of storage and fast transfer rates. ECS can be used only for program and data storage, not for program execution. Special hardware instructions exist for transferring data between central memory and ECS.

Extended Memory

An extension to central memory which is physically located outside of the machine. Also referred to as extended core storage (ECS), extended semiconductor memory (ESM), large central memory (LCM), or unified extended memory (UEM).

Family Device

Mass storage permanent file device associated with a specific system. A family may consist of 1 to 63 logical devices. Normally, a system runs with one family of permanent file devices available. However, additional families may be introduced during normal operation. This enables users associated with the additional families to access their permanent files via the alternate family.

Family Name

A designation that the installation may give to a group of permanent file devices.

Family Ordinal

An index into the family ordinal table (FOT). The family ordinal is used to identify a unique family.

Family Ordinal Table (FOT)

A central memory resident table used to associate family names with family ordinals.

Field Length (FL)

The area in central memory allocated to a particular job. The only part of central memory that a job can directly access.

File

1. A set of information that begins at beginning-of-information (BOI), ends with end-of-information (EOI), and can be referenced by a local file name.

2. That portion of a multiple file terminated by an end-of-file (EOF).

3. Data recorded on a magnetic tape beginning after HDR1 label and ending before an EOF1 label.

FOT

Refer to Family Ordinal Table.

IAF

Refer to Interactive Facility.

Indirect Access File

A NOS permanent file that you access by making a temporary copy of the file (GET or OLD commands). You create or alter it by saving or substituting the contents of an existing temporary file (REPLACE or SAVE commands). Compare with Direct Access File.

Interactive Facility (IAF)

An application that provides a terminal operator with interactive processing capability. The interactive facility makes terminal input/output and file input/output appear the same to an executing program.

Interactive Origin Job

A job initiated from an interactive (time-sharing) terminal.

Job Sequence Name (JSN)

The unique system-defined name assigned to every executing job or queued file. The JSN is a string of 4 alphanumeric characters, or, if the job is a subsystem, 3 alphanumeric characters.

Job Status

A job attribute kept in the job's executing job table (EJT) entry. It is used by the system to determine if a job is rolled in or rolled out. If a job is rolled out, job status indicates why it was rolled out.

JSN

Refer to Job Sequence Name.

LCF

Refer to Local Configuration File.

LCN

Refer to Loosely Coupled Network.

LID

Refer to Logical Identifier.

Local Configuration File (LCF)

A file in the host computer system containing information on the logical makeup of the communication elements of the host. The file contains a list of the application programs available for execution in the host computer, and the users that can access it. This is a NOS direct access permanent file.

Local NPU

An NPU that is connected to the host via a coupler. A local NPU always contains a host interface program (HIP) for processing block protocol transfers across host/local NPU interface.

Logical Identifier (LID)

A 3-character alphanumeric string used to identify a particular mainframe. LIDs are identified by your site.

Login

The procedure used to gain access to the system.

Logout

The procedure used to end a terminal session.

Loopback

A software capability by which data flow from a mainframe to the network "loops back" to the mainframe. This allows testing on a single mainframe with one network access device (NAD).

Loosely Coupled Network (LCN)

A network of physically connected computer systems. The LCN environment allows jobs, data files, and messages to be transmitted from one computer system to another.

Machine Identifier (MID)

A 2-character identifier used to associate a specific machine with its access to a shared device.

MAG

Magnetic tape subsystem.

Mass Storage

The equipment used to hold temporary and permanent files within the system.

Mass Storage Device

An extended memory or disk unit which has defined logical attributes such as family, file residency, and so on.

Mass Storage Table (MST)

Table that contains an entry for each logical device in the configuration of mass storage devices currently available to the system.

MID

Refer to Machine Identifier.

MST

Refer to Mass Storage Table.

Multimainframe System

Network of physically and logically connected computer systems.

NAD

Refer to Network Access Device.

NAM

Refer to Network Access Method.

NCF

Refer to Network Configuration File.

NDL

Refer to Network Definition Language.

Network

An interconnected set of network elements consisting of a host and one or more NPUs and terminals.

Network Access Device (NAD)

The primary element in a loosely coupled network. Each NAD connects a computer system to the network.

Network Access Method (NAM)

A software product that provides a generalized method of using a communications network for switching, buffering, queuing, and transmitting data. NAM is a set of interface routines used by a terminal servicing facility for shared access to a network of terminals and other applications, so that the facility program does not need to support the physical structures and protocols of a private communications network.

Network Configuration File (NCF)

A network definition file in the host computer. This file contains information on network elements and permissible linkages between them. The status of the elements described in this file is modified by the NPU operator in the course of managing the network. This is a NOS direct access permanent file.

Network Definition Language (NDL)

The compiler-level language used to define the network configuration file and local configuration file contents.

Network Processing Unit (NPU)

The collection of hardware and software that switches, buffers, and transmits data between terminals and host computers.

Network Supervisor (NS)

A portion of the network software, written as a network access method (NAM) application program, which dumps and loads network processing units (NPUs) upon request.

Network Validation Facility (NVF)

A portion of the network software, written as a NAM application program, which performs application validation and all connection validation processing and supports login dialogue with the terminal user.

NPU

Refer to Network Processing Unit.

NS

Refer to Network Supervisor.

NVF

Refer to Network Validation Facility.

Order-Dependent

Used to describe items which must appear in a specific order.

Order-Independent

Used to describe items which need not appear in any specific order. Parameters, particularly those with keywords, may be order-independent.

Origin Type

A job attribute that indicates how a job entered the system. The four origin types are interactive origin, batch origin, remote batch origin, and system origin.

OUTPUT

The system-defined file which, by default, contains all the ouput from job processing. It is also known as the print or punch file.

Peripheral Microcode

Special type of software that resides in a peripheral controller. The controlware defines the functional characteristics of the controller.

Peripheral Processor (PP)

The hardware unit within the host computer that performs physical input and output through the computer's data channels.

Peripheral Processor Unit (PPU)

First-level peripheral processor (FLPP). A PPU is contained in the mainframe in a multimainframe environment and operates synchronously with the mainframe.

Permanent File

A mass storage file that is catalogued by the system so that its location and identification are always known to the system. They are protected by the system from unauthorized access according to privacy controls specified when they are created.

Physical Identifier (PID)

The unique 3-character identifier of a specific host.

PID

Refer to Physical Identifier.

PP

Refer to Peripheral Processor.

PPU

Refer to Peripheral Processor Unit.

Procedure

A user-defined set of instructions that can be referenced by name. The instructions consist of procedure directives and system commands.

Procedure File

A file containing one or more procedures.

Queue Priority

An attribute associated with input and output files. If all other factors are equal, queue priority is used to select the best file for processing.

Queued File

An input, print, plot, or punch file that has an entry in the queued file table (QFT). It is not assigned an EJT entry and is waiting to be selected for processing.

QFT

Refer to Queued File Table.

Queued File Table (QFT)

A central memory resident table containing a 4-word entry for all active input and output queue files.

RBF

Refer to Remote Batch Facility.

Remote Batch Facility (RBF)

A network application that provides a terminal operator with remote batch processing capabilities. RBF transfers input and output files between remote batch devices and NOS.

Remote Batch Job

A job submitted from a remote batch terminal.

Remote Host Facility (RHF)

A central processor program that executes at a system control point. It performs data buffering and switching and is the intermediary between application programs and the network.

Remote NPU

A network processing unit linked to a host computer through other network processing units.

RHF

Refer to Remote Host Facility.

Rollout

The removal of jobs from central memory to mass storage before execution is complete, so the control point and central memory can be assigned to another job. A job is rolled out when it is waiting for an external event, when its control point is needed by a higher priority job, or when it exceeds its central memory time slice.

Rollout File

A file containing a job (and system information) that has been temporarily removed from the main processing area of the system.

Scheduling Priority

An attribute associated with an executing job available for job scheduling. Scheduling priority is used to select the best executing service class job for processing.

SC

Refer to Service Class.

Service Class (SC)

An attribute associated with a queued file or executing job. Service class determines how the system services the job.

Store and Forward

A software capability by which data flows from a network through a mainframe (store and forward mode) and into another network.

Suspended Job

An interactive job placed in an inactive state. Processing is stopped immediately and recovery information is copied to the rollout file. Processing is resumed as if no interruption took place, if the job's EJT entry is recovered.

SYSLIB

Refer to System Library.

System Job

A job brought to a control point by the operator.

System Library (SYSLIB)

The collection of tables and object language programs residing in central memory or on mass storage which are necessary for running the operating system and its product set.

System Origin Job

A job entered at the system console.

TCU

Refer to Trunk Control Unit.

Trunk

The communication line connecting two network processing units.

Trunk Control Unit (TCU)

The hardware part of a network access device (NAD) that interfaces with a network trunk.

UJN

Refer to User Job Name.

Unit Number

The setting of a hardware device. Used when more than one hardware unit can be connected to a controller.

User Job Name (UJN)

A 1- to 7-character alphanumeric name you specify to replace the system-defined job sequence name (JSN) for a queued file or executing job.

Volume Serial Number (VSN)

A 1- to 6-character identifier that identifies the volume of magnetic tape to the system.

VSN

Refer to Volume Serial Number.

Warmstart

Procedure used to deadstart if the tape or disk controller is loaded and the controlware is running.

INSTALLATION COMMANDS, PARAMETERS, AND PROCEDURES B

The following commands, parameters, and procedures used during installation are detailed in this appendix:

- COMMOD file parameters
- DECKLIS procedure
- Disk installations
- GENDST procedure
- MISCGET procedure
- REPORT procedure
- RESETP procedure
- SETUP procedure
- SYSGEN command

COMMOD FILE PARAMETERS

The installation procedures in DECKOPL contain two types of parameters: those that are unique to a specific installation procedure and those that are common to all installation procedures.

DECKOPL is released with a default value set for all of the common parameters. These defaults are in a modification set file named COMMOD. When you use the SETUP procedure to apply the modification set in file COMMOD against DECKOPL, SETUP sets all the default parameters in the INSTALL procedure file.

The COMMOD file is created automatically by the SYSGEN(SOURCE) function. Also refer to SYSGEN(SOURCE) and COMMOD Procedure in this appendix.

NOTE

The COMMOD file parameters you must change to perform a disk installation are described later in this appendix.

If you want to change the values for any of these parameters, follow these steps:

1. Edit the file COMMOD using an available editor.

2. Change the parameters values for your site.

3. Replace file COMMOD.

4. Use the SETUP procedure to build a new INSTALL file from DECKOPL that contains your modifications to COMMOD. For example:

   BEGIN,SETUP,INSTALL,MOD=COMMOD,INSTALL.

The following pages describe the parameters in file COMMOD.

D=option (called from common deck COMDEN)

Specifies the tape density. The default value is PE.

HI	556 characters per inch (cpi) (7-track).
HY	800 cpi (7-track).
HD	800 cpi (9-track).
PE	1600 cpi (9-track).
GE	6250 cpi (9-track).

NOTE

This parameter only affects tape requests for deadstart tape and density for output PLs when DISKINS=NO and OUTPRD=YES.

DISKINS (called from common deck COMDISK)

Controls the type of installation. The default is DISKINS=NO.

NO	Magnetic tape installation. This option keeps as many files on tape as possible. Only the composite OPL and the CCP/CROSS permanent files are kept on disk.
YES	Disk installation; if you want to use auxiliary disk packs, additional parameters relating to disk pack name and type must be changed from the zero defaults. Refer to Disk Installations, later in this appendix.

IA (called from common deck COMIA)

Specifies how an installation job is submitted for execution. If you do not specify the keyword IA, the installation procedure submits a batch job for execution except if the job origin type is BCOT (local batch). For batch origin jobs, the build procedure does not submit another batch job.

If the job origin type is not batch, you can specify the keyword IA to cause immediate execution of a build job. Specifying IA also causes the following:

- If the job origin type is interactive (as opposed to running under DIS), the file OUTPUT is assigned to mass storage and at job completion or abnormal termination is renamed IAOUT.

  If you want to assign the output back to the user's terminal rather than mass storage, enter this command:

  ```
  ASSIGN,TT,OUTPUT.
  ```

- The build procedure issues a LIBRARY,GLOBAL command when the job is finished executing. This causes the local file GLOBAL (if one exists) to be made the global library. This feature allows you to have your own global library named GLOBAL in effect at a terminal to run build jobs, and to have GLOBAL remain in effect after the build jobs are completed.

ICHG (called from common deck COMIUN)

Specifies the valid charge and project number values for the installation user name. Here is a sample value that causes the command CHARGE(1187,594N321) to be executed:

```
*N=$1187,594N321$.
```

IFAMILY (called from common deck COMIUN)

Specifies the default value for the alternative family name. Set this value if you are not installing on the system default family. This parameter is significant only if IUCHG=YES.

IPW (called from common deck COMIUN)

Specifies the password for IUN (installation user name). The default password is INSTALL. The password is used by SUBPROC. This parameter is significant only if IUCHG=YES.

IUCHG (called from common deck COMIUN)

Controls the source of USER and CHARGE commands for batch job submittal. The default value is YES.

YES	The parameter values specified by IUN, ICHG, IPW, RESOUR, and PACKNAM are used to generate a USERCHG file.
NO	You must create a USERCHG file (containing appropriate USER, CHARGE, RESOURC, and PACKNAM commands) for the build jobs. The file can be local, direct, or indirect.

IUN (called from common deck COMIUN)

Specifies the installation user name. The default installation user name is INSTALL. This parameter is significant only if IUCHG=YES.

LIST=filename (called from common deck COMLIST)

Specifies the file for assembly or compilation listing output. If filename is OUTPUT, the listing is printed with the installation listing. If you specify any other filename, use the TOLIST parameter to specify the destination for the output file. Also, the file cannot be a magnetic tape file if the installation procedure uses the FORTRAN compiler.

NOTE

A different LIST parameter, called from deck COMCCPL, is used for CCP build procedures. Refer to section 5 for information about CCP/CROSS installation parameters.

OUTPRD (called from common deck COMDISK)

Controls the production of output program libraries in installation jobs. The default is NO.

YES — The build jobs write output program libraries. This does not apply to MODOPL or CCP. If OUTPRD=YES and DISKINS=NO, output program libraries are written to a RECLAIM dump tape with VSN=OUTPLS. All output program libraries are written to this tape (or multiple tapes with this volume serial number [VSN]).

NO — Output program libraries are not generated or written.

NOTE

If you do not apply user code to change program library common decks for those products also used as auxiliary PLs and you do not want to write output PLs, you can use the following parameter settings:

```
OUTPRD=NO
PSRIN=nnn
PSROUT=nnn
```

nnn defaults to the current release level. The input PLs are then used as auxiliary PLs and no product output files are written. Binaries are still stored in file PRODUCT. The released DECKOPL has the parameters set in this manner.

PCKNAM (called from common deck COMIUN)

Specifies the pack name and type if all files used in the installation process are to reside on an auxiliary disk pack. The entry format is (*N=$PACKNAM,pname/R=typr.$). The default is *. This parameter is significant only if IUCHG=YES.

PRID (called from common deck COMPRID)

Specifies the printer identification for routing print files from verification jobs.

PSRIN (called from common deck COMIN)

Specifies the 3-digit number identifying the PSR level of this release. The default is the current release level.

PSROUT (called from common deck COMOUT)

Specifies the 3-digit number identifying the PSR level which is appended to the requested file name for the output PL files. The default is the current release level. If you do not change this value and OUTPRD=YES, the output PLs will overwrite the input PLs.

RESOUR (called from common deck COMIUN)

Specifies the format of the RESOURC command to use; for example, RESOURC(DJ=1,GE=1). This parameter is significant only if IUCHG=YES.

TOBLD=option (called from common deck COMTOB)

Specifies the build listing disposition and determines whether the job OUTPUT goes to the wait queue. The default is TOBLD=NO.

YES — Job output (everything on file OUTPUT) is placed in the wait queue with a user job name the same as the installation procedure name, followed by the letter B (for example, AAM2B). If the procedure name is 7 characters, the last character is replaced by B.

NO Job output is printed at the central site, which is also the disposition if the job fails.

NOTE

The procedure JOBPASS and JOBFAIL contain inactive code that suggests how to handle files in the wait queue.

TOLIST=option (called from common deck COMTOL)

Specifies the assembly list file routing and whether file LIST goes to the wait queue. The default is TOLIST=NO; if the job fails, the list file is discarded.

YES — The file named in the LIST=filename parameter is routed to the wait queue; the user job name is the same as the installation procedure name, followed by the letter L (for example, AAM2L). If the procedure name is 7 characters, the last character is replaced by L.

NO — The list file, if defined, is local to the job and is discarded when the job terminates.

NOTE

When you route listings to the wait queue, these files are counted in the total number of jobs validated for your user name. Also, if you want to specify different options for the LIST, TOBLD, and TOLIST functions, you can recode the procedures named JOBPASS and JOBFAIL in DECKOPL. However, procedures do not apply to CROSS and CCP installations.

UN1 (called from common deck COMDISK)

Specifies the alternative user name for input source files. Set UN1 to a value only if you are rebuilding under a user name other than IUN. The default value is 0; it specifies that the user name under which the job is executing is used.

NOTE

If you specify a user name for UN1, DISKINS must be set to NO if the source program libraries have not been preloaded.

UN2 (called from common deck COMDISK)

Specifies the alternative user name for output source files. The default is to store the files using the same user name as the job is executing under.

NOTE

If you specify a user name with the UN2 parameter, OUTPRD must be set to NO.

VFYTAPE (called from common deck COMDEN)

This parameter only affects copying deadstart tapes. Specifies verifications of all tape transfers; default is verification. Use the following setting to eliminate verification:

(*N=)

DECKLIS PROCEDURE

The DECKLIS procedure lists the installation and support procedures on DECKOPL. Here is the format for the DECKLIS procedure call:

```
BEGIN,DECKLIS,INSTALL,param,param,...,param.
```

There are two types of parameters: keyword=value and keyword.

KEYWORD=VALUE PARAMETERS

NOTE

Include all the keyword=value parameters before using the keyword parameters.

REP=n	Specifies the number of additional copies to be printed. The default is 0.
MOD=fn	Specifies the name of a modification set file to be added to the PL for listing purposes. The file can be a local file or a direct or indirect access permanent file. The PL is not permanently updated.
EXPAND=nn	Specifies whether common deck calls will be expanded. You can specify YES or NO. The default is YES.

KEYWORD PARAMETERS

MODIF	Includes the default Modify list options on the listing.
UMODE	Lists only modified decks; used in conjunction with the MOD=fn parameter.
NODATA	Suppresses expansion of data sections for verification jobs.

Here are some examples of calls to the DECKLIS procedure:

```
BEGIN,DECKLIS,INSTALL,MOD=COMMOD,UMODE.

BEGIN,DECKLIS,INSTALL,NODATA.
```

DISK INSTALLATIONS

If you want to perform your installation with the source program libraries on disk allowing the installation procedures to manage disk files for you, change the DISKINS parameter in file COMMOD to YES.

You can preload all the source files to disk before beginning this process, or you can let the installation procedures do it for you. If you want to preload your files, refer to Using PFLOAD, later in this subsection.

Each build job calls the procedure PRDIN. At the first reference to the product release file, PRDIN looks for a local file of the correct name (for example, NAM5). If the file is not local, PRDIN attempts to attach the file from disk. When the file is not on disk, PRDIN issues a RECLAIM command to load the file from the permanent file tapes and then store the file on disk.

The disk files are given the name of the product source file concatenated with the value of PSRIN. For example, the disk file for NAM5 would be named NAM5630 (PSRIN defaults to the current release level number). Output program libraries are given the name of the product concatenated with the value of PSROUT.

By default, output program libraries are not written and the value of PSRIN is the same as the value of PSROUT. That value is the current release level number. Because of this naming convention, if you want to write output program libraries, you must change the value of PSROUT so that it is different from the value of PSRIN.

The parameters PSRIN and PSROUT are common parameters. Refer to COMMOD File Parameters, earlier in this appendix.

USING PFLOAD

The quickest way to preload your files is to use PFLOAD. The permanent file tapes were written by the RECLAIM utility which produces a tape format compatible with PFDUMP and PFLOAD. However, when you are using PFLOAD to preload your files, note the following:

- You must specify the DI parameter to force PFLOAD to load files to the proper user index.
- You must specify the FM and DD parameters if you are loading a family device.
- You must specify the PN parameter if you are loading an auxiliary device.
- If you are loading multiple disk packs, you might want to use the OP=L option to cause PFLOAD to perform load leveling.

All source program libraries are loaded with names of this format:

xxxxpsrin

xxxx	The 4-character product name; for example, TEXT and NAM5.
psrin	The value of the PSRIN parameter which, by default, is the current release level number.

Thus, the source program library for NAM5 would be loaded with the name NAM5630. The name for the composite OPL is OPLpsrin. Thus, for this release, the name would be OPL630.

All other files are given names using this format:

PFGxxxx

xxxx The 4-character product name.

The PFG files are used by SYSGEN. Refer to SYSGEN command, later in this appendix, if you want to have SYSGEN use these disk files or if you want to purge the files and have SYSGEN use the files from tape.

DISK INSTALLATION WITH AUXILIARY PACKS

This type of installation uses the same steps as normal disk installation. However, you must change the parameter defaults on the common decks that contain the pack name and type definitions. The released defaults are zero. Each product source file is assigned to one of the common decks named COMD1A, COMD1B, COMD2A, COMD2B, or COMD3A. The parameters in these decks define the pack name and type for the disk pack location of the input and output source files. Normal and auxiliary program library assignments are listed in table B-1.

Each of the common decks contain four parameters. Use the first two parameters to define the disk pack name and type for storage of the input source file. Use the other two parameters to define the assignment for the output source file which you can assign to a different disk pack.

Each product source file that also serves as an auxiliary program library requires a second common deck defining its disk pack location. The second common deck is used in build jobs that use the auxiliary program library. This second deck describes the auxiliary program library disk location. It is named using the following format:

COMabcc

COM	Identifies a common deck on DECKOPL.
a	A unique digit or letter that associates this common deck with its product.
b	Is always given the value of X to identify its auxiliary program library use, except for COMSOPL.
cc	Two characters, identical to the last two characters of the common deck used by PRDIN to locate the input product file.

Example:

The RHC product file has its disk assignment in COMD1A. Product RHF uses RHC as an auxiliary program library and, thus, includes the deck COM6X1A. The pack assignment could be as follows:

```
COMD1A       COM6X1A

PN=PACKY
PR=DJ
PNO=PACKX    PN6=PACKX
PRO=DJ       PR6=DJ
```

Table B-1. Source File and Auxiliary PL Assignments

Common Deck	Source File	Auxiliary Common Deck†	Product Job	Required AUXPLs††	Common Deck	Source File	Auxiliary Common Deck†	Product Job	Required AUXPLs††
COMD1A	DDL3	COM4X1A	DDL3	1	COMD1B	LDR1		LOADER	
COMD1A	FTI4		FTNTS	1	COMD1B	AAM2	COM3X1B	AAM2	
COMD1A	FTN5		FTN5	1,5				TOOLS	S
COMD1A	SRT4		SORT		COMD2A	CID1		CID	S
COMD1A	FCL5		FCL5	1	COMD2A	SYMP		SYMPL	
COMD1A	SRT5		SORT5		COMD2A	DCL2		DCL	
COMD1B	PMD5		PMD	1	COMD2A	AGL5		ALGOL5	
COMD1A	CDCS		CDCS2	S,1,3,4	COMD2B	RBF5		RBF5	S,7
COMD1A	ITF1		ITF	6,S	COMD2A	NSS1		NSS	S
COMD1A	PASC		PASCAL	S	COMD2A	FMAT		FORMAT	S
COMD1A	IMF1		IMF1	1,4					
					COMD2A	MDD1		MDD	S
COMD1A	RHF1		RHF	S,6	COMD2B	MCS1		MCS	S,7
COMD1A	RHP1		RHP	S,6	COMD2B	ZMTS		MTS	C,S
COMD1A	RHC1	COM6X1A	RHC						
COMD1A	MSSI		MSSI	S					
COMD1A	MMCL		MMCL						
COMD1A	AP1I		AP1I						
COMD1B	TDU1		TDU						
COMD1B	TXIO		TEXTIO		COMD2B	CEDG		CEDIAG	S
COMD1B	BAS3		BASIC3	S,1	COMD2B	NAM5	COM7X2B	NAM5	S
COMD1B	ZHCD		HCD		COMD2B	COB5		COBOL5	
COMD1B	BIT8		BIT8		COMD2B	FCS3		FCS3	
COMD1B	TEXT	COM2X1B	TEXT		COMD2B	APL2		APL2	S
COMD1B	FDBF		FDBF	1,4	COMD2B	LCS3		LCS3	
COMD1B	FTN4		FTN4	1				HSIO	S
								FSE	S
COMD1B	PLI1		PLI	1,5				MMF	S
COMD1B	FCL4		FCL1	1				MSS	S
COMD1B	CCG1	COM5X1B	CCG					TAF	S
COMD1B	BINE		BINEDIT	1				IAF	S
COMD1B	FORM		FORM					NOS	S
COMD1B	UPD1		UPDATE	1				TRACER	S
COMD1B	F451		F45	1				XEDIT	S
COMD1B	FCL4		FCL2					MAS	S
COMD1B	QU31		QU3	S,4				NIP5870	S
COMD1B	CPS1	COM1X1B	COMPASS						
COMD1B	CCL1		CCL	1,S					
COMD1B	BAM1		BAM	2					
COMD1A	TLIB		OSTEXT	S					
COMD1A	TLIB		OSLIB	S					
COMD1A	NOSB		NOS2B	S					
COMD1A	NOSB		RDFEX	S					

†A common deck name in this column identifies the source file as one also used as an auxiliary PL.
††Each letter or digit in this column refers to common decks for auxiliary PLs needed by this build job. For example, S means COMSOPL, 1 means COM1X1B, 2 means COM2X1B, and C means COMCW (controlware).

GENDST PROCEDURE

Use the GENDST procedure to merge the binaries on file PRODUCT with a base deadstart file and generate a new deadstart file. The GENDST procedure ensures that there are no conflicts between IAF and RDF on the new deadstart file.

Here is the format of the GENDST call:

```
BEGIN,GENDST,INSTALL,SYSTEM=odt,NEW=ndt,D=density,LIST=list.
```

Parameter	Description
SYSTEM=odt	Local file name for the old deadstart file. If file odt is preassigned, it becomes the base deadstart file; otherwise, a tape label request is issued with a VSN=ODT. The default file name is SYSTEM.
NEW=ndt	Local file name for the new deadstart file. If file ndt is preassigned, the new deadstart file is written on it; otherwise, a tape label request is issued with a VSN=NDT. The default file name is NEW.
D=density	Tape density option. If this parameter is omitted, the value in deck COMDEN set in COMMOD is used. The default is PE. The option applies to both odt and ndt. Here are the values you can specify for tape density:

density	Significance
HI	556 cpi (7-track).
HY	800 cpi (7-track).
HD	800 cpi (9-track).
PE	1600 cpi (9-track).
GE	6250 cpi (9-track).

Parameter	Description
LIST=list	Local file name for the listing file. The default file name is OUTPUT. If you run GENDST interactively and want to save the listing file or do not want it to appear at the terminal, specify a different file name.

Run a job similar to the following to add site-provided binaries and deadstart decks to the new deadstart file. Create file USERD so it contains the LIBEDIT directives (refer to NOS 2 Reference Set, Volume 3) to make these additions.

Job	Comments
job command. USER,username,password,familyname. GET,USERD.	USERD contains the LIBEDIT directives.
GET,lfn=pfn.	lfn (permanent file name is pfn) contains the modified deadstart decks. lfn must appear in the USERD file as *FILE lfn.
BEGIN,GENDST,INSTALL.	Parameters are not required if you use the system defaults.

MISCGET PROCEDURE

All code that may correct a specific user site problem but which has not been fully tested is contained on the file MISCPL. This file was created during the setup of installation files. The modifications are properly formatted (Update or Modify) for the intended program library.

To list the modification set headers and the decks containing calls to common decks containing modification sets, use the following commands:

```
BEGIN,MISCGET,INSTALL,HISTORY.
ROUTE,USER,DC=PR.
```

You can extract code from MISCPL using any of the following methods.

- To extract all modification sets for a product, use the following command:

  ```
  BEGIN,MISCGET,INSTALL,PRD=name.
  ```

  Replace name with the name for the deck containing a call for each modification set available for the product.

- To extract one modification set, use the following command:

  ```
  BEGIN,MISCGET,INSTALL,MOD=modset.
  ```

  Replace modset with the name of the required modification set.

- To extract selected modification sets, use the following commands:

  ```
  NOTE,lfn,NR.+,modname+,modname+,.....,modname.
  BEGIN,MISCGET,INSTALL,MISCIN=lfn.
  ```

These calls to MISCGET append the modification sets to the local file USER. You should then save file USER as a permanent file for later use by the corresponding build procedure.

REPORT PROCEDURE

To obtain statistics on all completed installation jobs, run a job that contains this command. (The job output indicates the resources used for each installation job and whether the job passed or failed.)

```
BEGIN,REPORT,INSTALL,XC.
```

NOTE

The REPORT procedure uses the direct access file REP. If the binary file cannot be found or if the XC keyword is present, REP is recompiled.

RESETP PROCEDURE

You can use the RESETP procedure to reduce the size of files PRODUCT and DIRFILE. This procedure makes more disk space available and speeds up the subsequent LIBEDITs of more binaries into file PRODUCT.

Initially, procedure SEED creates the file PRODUCT with user libraries that are used by the build procedures, and creates DIRFILE with entries for those ULIBs. The build procedures subsequently add binaries to the file PRODUCT and directives to file DIRFILE. Only those user libraries used by subsequent build jobs are required to remain on file PRODUCT after GENDST creates a new deadstart tape.

To use the RESETP procedure to reduce the size of files PRODUCT and DIRFILE, follow these steps:

1. Run GENDST to create a new deadstart tape.

2. After writing the new deadstart tape, enter this command:

   ```
   BEGIN,RESETP,INSTALL.
   ```

SETUP PROCEDURE

The SETUP procedure generates the permanent file INSTALL, which contains all installation procedures. The SYSGEN(FULL) function (as well as other SYSGEN functions) initially loads files from the permanent file tapes and creates INSTALL. SETUP can also perform the following functions:

- Create INSTALL from DECKOPL with optional modifications against DECKOPL.
- Rename file INSTALL to a specified name.
- Create a 63-character set version of procedures on INSTALL.
- Convert DECKOPL and INSTALL to a 63-character set format.
- Replace DECKOPL with an updated DECKOPL.

Here is the format for calling SETUP:

```
BEGIN,SETUP,INSTALL,params.
```

Parameter	Description
NEWPL	Replaces DECKOPL with modified DECKOPL. If NEWPL is omitted, DECKOPL is not replaced.
MOD=filename	Applies modsets from specified file to DECKOPL; the file can be a local file or a direct access permanent file. If you change any default parameters in COMMOD and you also have your own local changes to DECKOPL, append your changes to file COMMOD. Then use MOD=COMMOD in your procedure call.
DF63	Selects 63-character-set version of installation procedures for inclusion on file INSTALL.
CV63	Converts DECKOPL to 63-character-set format.
INSTALL=filename	Creates or replaces procedure file specified. The default is INSTALL. If you omit the INSTALL keyword, procedure file INSTALL is not created or replaced.

Example:

The following sample call to the SETUP procedure applies modifications from file COMMOD against the DECKOPL installation procedures and creates a new INSTALL procedure file. DECKOPL is not updated.

```
BEGIN,SETUP,INSTALL,MOD=COMMOD,INSTALL.
```

SYSGEN COMMAND

SYSGEN consists of several procedures for loading all the permanent files for NOS and its product set from the released permanent file tapes. SYSGEN also contains procedures to perform the following:

- Create a new deadstart tape; refer to SYSGEN(DST).
- Merge binaries received with a component order with an existing deadstart tape; refer to SYSGEN(UPGRADE).
- Load files needed to perform a customized installation; refer to SYSGEN(SOURCE).
- Copy the running system to tape or disk; refer to SYSGEN(COPYSYS).
- Move files from the installation user name INSTALL to a system user name; refer to SYSGEN(MOVE).
- Make a copy of the release materials; refer to SYSGEN(COPYTAP).
- Build a new deadstart tape with the trace/debug versions of NAM, RBF, and/or PSU; refer to SYSGEN(SWAP).
- Add a 63-character set version of PSU to your system.

SYSGEN VALIDATIONS

To use SYSGEN, your validation file must contain the correct user names and passwords. If you do not have a validation file, the SYSGEN(FULL) function automatically creates one with the necessary validations. If you have a validation file, modify the file via MODVAL so that it contains the user names and passwords that SYSGEN needs. If you need to modify SYSGEN, refer to SYSGEN maintenance in this section. Then, after completing your installation, restore the file for security. Table B-2 shows a list of the user names and passwords for SYSGEN.

Table B-2. SYSGEN User Names and Passwords

User Name	Password	Files Installed
LIBRARY	LIBRARY	FSE, XEDIT, and CCL (command help) files.
SYSTEMX	SYSTEMX	NAM and other subsystem startup procedure files.
NETOPS	NETOPSX	NAM and related subsystem procedure startup files and CCP load files.
MANUALS	MANUALS	Online manual files.
INSTALL	INSTALL	RECLAIM database file, DECKOPL, and other files needed for installation jobs.
APL0	APL0	For APL2 product only.
APL1	APL1	For APL2 product only.
SSPOT	STATION	For NOS Scope Station Facility only; no files installed.
CDCCE	CDCCE	Customer engineer files for MSSI and MAP III or MAP IV-23/25.
CDCCE2	CDCCE2	Customer engineer files for MSSI and MAP IV-20/21.
KB100DC	TAFPASS	For the TAF task library, TASKLIB.
SUBFAM0	SUBFAM0	For the Mass Storage Extended configuration file, MSASCON.

SYSGEN MAINTENANCE

The SYSGEN procedures are maintained in deck SYSGEN on DECKOPL. To create a new set of SYSGEN procedures, execute this command:

```
BEGIN,SYSGEN,INSTALL.
```

The SYSGEN procedures consist of a PROC/SYSGEN and a ULIB/PFGLIB on the deadstart file. The BEGIN(SYSGEN,INSTALL) command adds SYSGEN to the PRODUCT and GLOBLIB files. (If GLOBLIB and PRODUCT are attached and a LIBRARY(GLOBLIB) has been done, all SYSGEN functions execute from PRODUCT and GLOBLIB from the global library.)

If you need to modify SYSGEN (for example, to set correct user names), you should append your modifications to file COMMOD. Then, run the SETUP procedure, using MOD=COMMOD, to create a new INSTALL procedure file.

All user names used by SYSGEN are stored in the USER procedure within deck SYSGEN. SYSGEN(MODVAL) is the routine that creates the user names.

SYSGEN FUNCTIONS

The SYSGEN functions set up all the permanent files for NOS and its product set. You can execute all SYSGEN functions from DIS, from an interactive terminal, or from a batch job.

If you are executing the SYSGEN functions from the system console, use the X.SYSGEN format call from DSD. You can also execute the SYSGEN functions by using these sequences of commands:

```
X.DIS.
SUI(userindex)
SYSGEN(function,params)

          or

X.DIS.
USER(username,password)
SYSGEN(function,params)
```

These sequences will install the files to the specified user name/user index. If you are executing SYSGEN from the interactive terminal, the files will be installed to the login user name. For example, you can execute the SYSGEN(FSEH) function from your terminal and install the FSE help files to your own permanent file catalog rather than the LIBRARY catalog. Thus, you can examine the files before you install them for the entire system.

Replace function with the name of the function you want to execute. The following pages detail all the SYSGEN functions in alphabetical order.

Replace params with the possible parameters for a specific function.

Some of the SYSGEN functions should be executed from DSD. To execute a function from DSD, type X. before the command. For example,

```
X.SYSGEN(FULL)
```

The SYSGEN functions that are run from DSD are documented with the X. before the command.

SYSGEN(COPYSYS)

This function copies a running system file to disk or tape. Use this format to call the function:

```
X.SYSGEN(COPYSYS,est,type)
```

Replace est with the EST ordinal of the device you want to copy to.

Replace type with either the word DISK for copying a running system to disk, or a value for a tape density: HY, HD, PE, or GE. If you are copying to disk, the procedure executes the INSTALL command to create a disk deadstart file. If you are copying to tape, the procedure executes the ASSIGN command. Thus, you should mount the tape you are copying to before executing this function.

SYSGEN(COPYTAP)

This function makes copies of the release materials: the deadstart tape, permanent file tapes, and 700 CIP tape. To use this function, you need two tape drives. Here is the format to call the function:

```
X.SYSGEN(COPYTAP,aaaaa,dl,numpf,dst,cip)
```

Parameter	Description
aaaaa	Specifies the first 5 characters of the VSN of the permanent file tape.
dl	Specifies the density of the release tapes.
numpf	Specifies the number of permanent file tapes to copy. Specify 0 if you do not want to copy any of the tapes.
dst	Specifies whether to copy the deadstart tape. Enter YES or NO.
cip	Specifies whether to copy the CIP tape. Enter YES or NO.

If you specify that permanent file tapes should be copied, the function blank-labels numpf tapes. Be sure to use the same size tapes as the release tapes. The function then requests the first original tape and the first new tape and begins copying. Mount the tapes when you are requested to do so.

After the permanent file tapes are copied, the function copies the deadstart tape and then the CIP tape.

SYSGEN(DST)

This function creates a new deadstart tape. Use this function when you have made simple changes to your deadstart file. For example, you have added or replaced deadstart decks. If you want to create a new tape using the PRODUCT file, use the GENDST command (refer to GENDST in this appendix). Here is the format for the SYSGEN(DST) function:

```
SYSGEN(DST,old,lgo,new,input,dd)
```

Parameter	Description
old	Name of the base deadstart file. This deadstart file should be local to your job or you should supply the dd parameter so that SYSGEN can request a deadstart tape to use as the base. If a tape is requested, the VSN is ODS. This file can be a preassigned magnetic tape file.
lgo	Name of the file which contains the binaries to update file old. If lgo is not local, SYSGEN looks for a permanent file with that name. You can specify 0 (zero) if no binaries are to be updated.
new	Name of file to receive the new deadstart file. This file can be a preassigned magnetic tape.

Parameter	Description
input	Name of a file containing directives for the LIBEDIT that SYSGEN performs. Specify 0 (zero) if you have no LIBEDIT directives.
dd	Tape density for any tapes to request. If you don't specify a tape density, no tape requests are made and SYSGEN only deals with local files.

SYSGEN(FILES)

Use this function to load all the permanent files from the permanent file tapes that are released with a component order. This function loads all the files from the permanent file tapes except for the RECLAIM database. Also, this function does not create a validation file. Use SYSGEN(FILES) after using the SYSGEN(UPGRADE) function when you are installing a component order.

Here is the format for calling SYSGEN(FILES):

```
X.SYSGEN(FILES,vsn)
```

Replace vsn with the VSN listed on the external tape label.

SYSGEN(FULL)

The SYSGEN(FULL) function loads all the permanent files for a standard installation. First, SYSGEN(FULL) loads your RECLAIM database and sets up a validation file (if you have none). Next, SYSGEN(FULL) loads the permanent files. The types of files loaded include all subsystem startup procedure files, help files, online manuals, and other permanent files associated with specific products. If you are performing a customized installation, refer to SYSGEN(SOURCE) in this subsection. Here is the format for calling SYSGEN(FULL):

```
X.SYSGEN(FULL)
```

SYSGEN(MOVE)

The SYSGEN(MOVE) function moves permanent files produced by installation jobs to other user names. (The user names must be known to SYSGEN. Refer to SYSGEN Maintenance earlier in this subsection.) You must execute SYSGEN(MOVE) from the system console. Here is the format for calling SYSGEN(MOVE):

```
X.SYSGEN(MOVE,pfn,un,ct,m,ac)
```

Parameter	Description
pfn	Name of the permanent file to be moved. The file must reside on installation user name INSTALL.
un	User name to receive files. Valid user names are INSTALL, SYSTEMX, LIBRARY, KB100DC, CDCCE, CDCCE2, MANUALS, SSPOT, NETOPS, APL0, APL1, and SUBFAM0. SYSGEN supplies the password. For the list of passwords, refer to SYSGEN Validations earlier in this subsection.
ct	Specifies the permanent file category. Allowable values are PU for public, PR for private, and S for semiprivate. The default is PR.
m	Specifies permission mode. Allowable values are R for read only and W for write. The default is R.
ac	Specifies the alternative catalog CATLIST option. You can enter Y to allow alternative catalogs to CATLIST the file, or you can enter N to prevent alternative catalogs from using CATLIST to list the file.

SYSGEN(RECLAIM)

This function loads files from the permanent file tape(s) and leaves them as local files. This function should not be called from DSD but can be called from DIS or an interactive terminal. Here is the format for the call:

```
SYSGEN(RECLAIM,f1,f2,f3,f4,f5)
```

Replace f1 through f5 with the names of the files you want to load from the dump tape(s). (You can load up to five files with each call.)

NOTE

To use SYSGEN(RECLAIM), you must have previously loaded the RECLAIM database for the dump tapes to user name INSTALL. (This is done automatically by the SYSGEN(FULL) or SYSGEN(SOURCE) function.)

SYSGEN(SOURCE)

This function loads files from the permanent file database and enables you to begin a customized installation. This function can be run from DSD, DIS, or from an interactive terminal. Refer to section 3 for more information. Here is the format for calling the function:

```
X.SYSGEN(SOURCE,CCP)
```

Include the keyword CCP if you want to begin a customized installation of CROSS/CCP at the CCPVAR step. (Refer to section 5.)

SYSGEN(SOURCE) sets up the RECLAIM database and the files DECKOPL, COMMOD, CODEPL, MISCPL, NAMSTRT, USERBPS, INSTALL, and DBUBIN. These are set up on the same user name as the SYSGEN(SOURCE) is executing from. SYSGEN(SOURCE) also puts a copy of RECLAIM on an indirect access permanent file.

Once SYSGEN(SOURCE) has completed, you can use RECLAIM to examine various release materials.

SYSGEN(SWAP)

This function creates a new deadstart tape with the DEBUG/TRACE versions of NAM, RBF, and PSU. Here is the format for calling this function:

```
X.SYSGEN(SWAP,density,p1,p2,p3,type)
```

Parameter	Description
density	Specifies the density of the new deadstart tape.
p1,p2,p3	Specify the products to be swapped. You can specify RBF, NAM, and PSU. Any combination can be used.
type	Specify the type of PSU to swap in. You can specify T64 for trace 64-character set; NT63 for nontrace 63-character set; or T63 for trace 63-character set. The default is T64. (This parameter is ignored for RBF and NAM.)

NOTE

Be sure to keep a copy of your original tape that contains the non-DEBUG/TRACE binaries of the products.

SYSGEN(UPGRADE)

The SYSGEN(UPGRADE) function merges the binaries you receive for a component order with your existing system. You can use SYSGEN(UPGRADE) to perform any of these functions:

- Merge the component-order binaries with a tape copy of your system and then create a new deadstart tape.
- Merge the component-order binaries with a local file copy of your system and then create a new deadstart local file. (You can then write the file to tape or create a disk deadstart file.)
- Update the RECLAIM database without merging the binaries.

To merge binaries with a tape copy of your system and create a new deadstart tape, use this format:

```
X.SYSGEN(UPGRADE,OLD,NEW,vsn,d1,d2)
```

To merge binaries with a local file copy of your system and create a new local deadstart file, use this format:

```
SYSGEN(UPGRADE,old,new,vsn,dl)
```

To update the RECLAIM database without merging the binaries, use this format:

```
X.SYSGEN(UPGRADE,0,0,vsn,dl)
```

Parameter	Description
old	Name of local file containing the binaries of the current system.
new	Name of local file to receive the merged binaries.
vsn	Volume serial number of the CDC-supplied permanent file dump tape. The VSN is listed in the media number field on the external tape label.
dl	Tape density of the permanent file dump tape.
d2	Tape density of both the old and new deadstart tapes. The old deadstart tape is requested with the VSN of ODT; the new deadstart tape is requested with the VSN of NDT.

FILE FORMATS C

ORDER OF PRODUCTS

The order in which products appear on the deadstart tape is now controlled by DECKOPL in each build deck. The basic order is alphabetical.

NOTE

All ULIBs are in library 4.

Arrangement of deadstart tape libraries:

Library	Description	Contributing Build Decks
1	Fixed order (CTI)	
2	Fixed order	NOS, HSIO, MMF, DISK CONTROLWARE
3	NOS	NOS, NOS2B, NIP5870
4	All ULIBs	Many
5	Miscellaneous items	TDU, PFTF, Misc. firmware
6	AAM2	AAM2
7	AGL5	ALGOL5
8	APL2	APL2
9	BAM1	BAM
10	BAS3	BASIC3
11	BINE	BINEDIT
12	BIT8	BIT8
13	CCL1	CCL
14	CDCS	CDCS2
15	CEDG	CEDIAG
16	CID1	CID
17	CML1	CML
18	COB5	COBOL5/COBOL5Q
19	CPS1	COMPASS
20	DDL3	DDL3
21	(empty)	
22	FDBF	FDBF
23	FORM	FORM
24	FMAT	FORMAT
25	FSE1	FSE
26	FTN4	FTN4/FTNTS,FCL1,FCL2
27	FTN5	FTN5,FCL5
28	F451	F45
29	IAF1/RDF1	IAF/RDFEX
30	IMF1	IMF1
31	ITF1	ITF
32	LDR1	LOADER
33	MMCL/MSSI	MMCL/MSSI/AP1I
34	MAS	MAS
35	MCS1	MCS
36	MDD1	MDD

Library	Description	Contributing Build Decks
37	MMF1	MMF
38	MSS1	MSS
39	NAM5	NAM5/NAM5D
40	(empty)	
41	NSS1	NSS
42	OSTX	OSTEXT
43	PASC	PASCAL
44	PLI1	PLI
45	PMD5	PMD
46	PSU1	PSU
47	QU31	QU3
48	RBF5	RBF5/RBF5D
49	RHF1	RHF
50	(empty)	
51	RHP1	RHP
52	SRT4	SORT
53	SRT5	SORT5
54	SYMP	SYMPL
55	TAF1	TAF
56	TEXT	TEXT
57	TXIO	TEXTIO
58	TOOL	TOOLS
59	TRCE	TRACER
60	UPD1	UPDATE
61	XEDT	XEDIT
62	(empty)	
63	(empty)	

The following is a list of all files on the permanent file tape(s). All source files are listed first, followed by all PFG files (files processed by SYSGEN). At the end of the list are files which are received only on a component order.

NOTES

Your tape contains only the files for the products you have ordered.

To see a list of all the files on your permanent file tape(s), follow these steps:

1. Load the RECLAIM database using either the SYSGEN(FULL) or SYSGEN(SOURCE) command.

2. Enter this command from your installation user name:

   RECLAIM,Z./LIST,UN=NS2psrin.

   Replace psrin with the release level, for example, 630.

This will generate a report that lists the names of all the files on the tape(s) in alphabetical order. Because the report is generated from information in the RECLAIM database, the permanent file tape(s) will not be requested.

SOURCE FILES

All source files end with the value psrin, the current NOS release level. For example, the AAM2 source file appears as AAM2630 at NOS level 630. The table does not list the psrin value.

Unless otherwise noted, all source files contain one random formatted UPDATE program library.

Source Files:

File Name	Associated DECKOPL Procedure Name	Format Notes
AAM2	AAM2	
AGL5	ALGOL5	
APL2	APL2	File 1. Program library. File 2. APLLIB. Relocatable of APL. File 3. APLPROD. Absolute of APL. File 4. TAPLTST. APL verification jobs. File 5. TAPLOUT. Sample output for file 4. File 6. NEWSF. APLNEWS, news file. File 7. FILESYS. Workspace, file functions. File 8. FILES2. Workspace, file functions. File 9. APLNEWS. Workspace, information. File 10. CATALOG. Workspace, information. File 11. WSFNS. Workspace, general functions. File 12. TAPLWS. Workspace for APL verification. File 13. Reserved.
AP1I	AP1I	AP1 Online Diagnostics for MAP III and MAP IV. File 1. FTN5 version of source PL. File 2. FTN4 version of source PL. File 3. CSTFU File 4. CSCMD File 5. CSQMM File 6. CSMAINT File 7. CSVDMT File 8. CS24KM File 9. CSDSOP4 File 10. CSECS File 11. CSECSTA File 12. CSHSMT for 24K Memory Size File 13. CSHSMT for 48K Memory Size File 14. CSHSMT for 64K Memory Size File 15. CSCMI for 64K Memory Size File 16. CSCMI for 48K Memory Size File 17. CSCMI for 24K Memory Size
BAM1	BAM	
BAS3	BASIC3	
BINE	BINEDIT	
BIT8	BIT8	
CCG1	CCG	
CCL1	CCL	

File Name	Associated DECKOPL Procedure Name	Format Notes
CCPB	CCPPH1, CCPBLB	File 1. Program library for CCP base.
		File 2. Expand and autolink binaries.
		File 3. Mux firmware (phase 1) load module.
		File 4. Dump bootstrap module.
		File 5. System autostart (SAM) load module.
		File 6. Symbol table for dump bootstrap load module.
		File 7. Combined binary library (BCMB) for generation of phase 2 variant load modules.
		File 8. Compiler listing for expand and autolink.
		File 9. CCP listing for phase 1 load module.
		File 10. CCP listing for dump bootstrap load module.
		File 11. CCP listing for system autostart load module.
		File 12. CCP assembly listing for BCMB.
		File 13. CCP Pascal listing for BCMB.
CCPD	CCPPH1	CCP online diagnostics - sequential PL.
CCPR	CCPPH1	CCP Remote Concentrator (LIP) - sequential PL.
CCPT	CCPPH1	CCP 3270 Bisync TIP - sequential PL.
CDCS	CDCS2	
CEDG	CEDIAG	
CID1	CID	
COB5	COBOL5	File 1. Sequential PL.
		Files 2,3,4. Used by the COBOL5Q build procedure.
CPS1	COMPASS	
CRSS	CROSS	File 1. Source program library.
		File 2. Absolute binary for Cross.
		File 3. Empty.
		File 4. Empty.
		File 5. Pascal cross-reference.
		File 6. MPLINK.
		File 7. MPEDIT.
		File 8. CCP Pascal compiler.
		File 9. Pascal compiler bootstrap.
		File 10. Empty.
		File 11. Cross program listing.
DCL2	DCL	File 1. Program library.
		File 2. SEGLOAD directives.
		File 3. Absolute for DCUPD.
		File 4. Absolute for DCSEL.
		File 5. Absolute for DCRPT.
		File 6. Absolute for DCRET.
		File 7. Absolute for DCCONVT.
		File 8. Absolute for DCUTL.
		File 9. Absolute for DCIDX.
		File 10. Absolute for DCGEN.
		File 11. Absolute for DCCONGN.
DDL3	DDL3	
FCL4	FCL1/FCL2	
FCL5	FCL5	

File Name	Associated DECKOPL Procedure Name	Format Notes
FCS3	FCS3	File 1. Program library. File 2. Binary data for FORTRAN file conversion processor (FCP) verification. File 3. Binary data for COBOL FCP verification. File 4. Absolute load module CBLFCP1 for COBOL FCP. File 5. Absolute load module CBLFCP2 for COBOL FCP. File 6. Absolute load module FTNFCP1 for FORTRAN FCP. File 7. Absolute load module FTNFCP2 for FORTRAN FCP. File 8. FORTRAN binary syntax file CBLFCPM for COBOL FCP. File 9. FORTRAN binary syntax file FTNFCPM for FORTRAN FCP.
FDBF	FDBF	
FMAT	FORMAT	
FORM	FORM	
FTI4	FTNTS	
FTN4	FTN4	
FTN5	FTN5	
F451	F45	
IMF1	IMF1	
ITF1	ITF	
LCS3	LCS3	File 1. Program library. File 2. Absolute load module for LCS. File 3. FORTRAN binary syntax for LCS. File 4. Absolute load module COUP for COSY-to-Update file conversion. File 5. Absolute load module COPYCOB for COBOL-COPY-library file conversion. File 6. Binary data file for COSY-to-Update file conversion (file 1 of 2). File 7. Binary data file for COSY-to-Update file conversion (file 2 of 2).
LDR1	LOADER	
MCS1	MCS	
MDD1	MDD	
MMCL	MMCL/AP1L	
MSSI	MSSI	
NAM5	NAM5	
NSS1	NSS	
OPL		This is a composite OPL of all the NOS MODIFY products you ordered. It may contain the PLs for NOS, NOS2B, FSE, HSIO, IAF, MMF, MSS, TAF, TOOLS, TRACER, MAS, RDFEX, and XEDIT.
PASC	PASCAL	
PLI1	PLI	
PMD5	PMD	
QU31	QU3	
RBF5	RBF5	
RHC1	RHC	

File Name	Associated DECKOPL Procedure Name	Format Notes
RHF1	RHF	
RHP1	RHP	
SRT4	SORT	
SRT5	SORT5	
SYMP	SYMPL	
TDU1	TDU	File 1. Modify PL TD7BASE. File 2. SESPLIB. File 3. TDUCOMM.
TEXT	TEXT	
TLIB	OSLIB	File 1. Contains TDU input file.
TXIO	TEXTIO	
UPD1	UPDATE	
VSM1	CCPVAR	CCP VARIANT VSM1 File 1. CCP variant (phase 2) load module. File 2. CCP PICB load module. File 3. Symbol table for CCP variant and PICB. File 4. CCP listing for CCP variant build (MPLINK). File 5. Listing of CCP variant build (MPEDIT). File 6. Listing of PICB build (Pascal, MPLINK, and MPEDIT).
VV1F	CCPVAR	CCP variant V1F; same format as file VSM1.
VV1L	CCPVAR	CCP variant V1L; same format as file VSM1.
VV2F	CCPVAR	CCP variant V2F; same format as file VSM1.
VV2L	CCPVAR	CCP variant V2L; same format as file VSM1.
VV3F	CCPVAR	CCP variant V3F; same format as file VSM1.
VV3L	CCPVAR	CCP variant V3L; same format as file VSM1.
VLC1	CCPVAR	CCP variant SM; same format as file VSM1.
VLC2	CCPVAR	CCP variant SM; same format as file VSM1.
VLC3	CCPVAR	CCP variant SM; same format as file VSM1.
VLC4	CCPVAR	CCP variant SM; same format as file VSM1.
ZHCD	HCD	819 PP Driver.

All of the following files are processed by SYSGEN. They are stored on the permanent file tape(s) and have this naming format:

PFGxxxx

where xxxx is a four-character product name (for example, APL2).

The following table lists each file name, the file's associated DECKOPL procedure name, the SYSGEN procedure which installs the file, and the format of the file. When a DECKOPL procedure is listed, it is the name of the procedure that generates the associated file. Many of the files listed do not have associated DECKOPL procedures. To install the files that do not have an associated DECKOPL procedure, you must either use the SYSGEN function or explicitly get the file from the tape by using the RECLAIM command or the SYSGEN(RECLAIM) function.

All files loaded by SYSGEN(SOURCE,CCP) are marked with an asterisk (*). All other files are loaded by SYSGEN(FULL) or SYSGEN(FILES,vsn).

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	SYSGEN Function/Notes
PFGAPL2	APL2	APL2	File 1. Empty file. File 2. APLLIB. Relocatable of APL. File 3. APLPROD. Absolute of APL. File 4. TAPLTST. APL verfication job. File 5. TAPLOUT. Sample output for file 4. File 6. NEWSF. APLNEWS, news file. File 7. FILESYS. Workspace, file functions. File 8. Files2. Workspace, file functions. File 9. APLNEWS. Workspace, information. File 10. CATALOG. Workspace, information. File 11. WSFNS. Workspace, general functions. File 12. TAPLWS. Workspace for APL verfication.
PFGAP1I	AP1I	AP1I	AP1 Online Diagnostics for MAP III and MAP IV. File 1. CSTFU. File 2. CSCMD. File 3. CSQMM. File 4. CSMAINT. File 5. CSVDMT. File 6. CS24KM. File 7. CSDSOP4. File 8. CSECS. File 9. CSECSTA. File 10. CSHSMT (24K memory). File 11. CSCMI (24K memory).

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	SYSGEN Function/Notes
PFGAP1L	AP1L	AP1L	MMCL files for AP1 Online Diagnostics. File 1. AP1LIB1; Channel Coupled MAP IV-2X. File 2. AP1LIB2; Channel Coupled MAP IV-2X. File 3. AP1LIB3; Channel Coupled MAP IV-2X. File 4. AP1LIB1; ECS/CMI Coupled MAP III/IV. File 5. AP1LIB2; ECS/CMI Coupled MAP III/IV. File 6. AP1LIB3; ECS/CMI Coupled MAP III/IV.
*PFGCCPB	CCPPH1/ CCPBLB	CCPB	File 1. Expand/autolink directives PL. File 2. Expand and autolink binaries. File 3. Mux firmware (phase 1) load module. File 4. Dump bootstrap module. File 5. System autostart (SAM) load module. File 6. Symbol table for dump bootstrap load module. File 7. Combined binary library (BCMB) for generation of Phase 2 variant modules. File 8. Updated combined program library (PCMB).
PFGCCPL	CCPLOAD	CCPL	File 1. Contains CCP NLFFILE.
*PFGCCPU		CCPU	File 1. Contains CCP USERBPS file.
PFGCDCS	CDCS2	CDCS	File 1. Contains CDCS startup prodedure.
*PFGCODE		CODE	File 1. Contains CODEPL.
*PFGCRSS	CROSS	CRSS	Cross Binary Install Permanent Files. File 1. Empty. File 2. Absolute binary for CROSS. File 3. Empty. File 4. Empty. File 5. Pascal cross-reference. File 7. MPEDIT. File 8. CCP Pascal compiler. File 9. PASCAL compiler bootstrap.
PFGCML1		CML1	Refer to CML Reference Manual for specific file format information.
*PFGCSTD		CSTD	File 1. COMPASS coding standards. File 2. SYMPL coding standards.
*PFGDBU1		DBU1	File 1. For use by QU3.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	SYSGEN Function/Notes
PFGDCL2	DCL	DCL2	Permanent files for Data Catalogue 2 V2.0. File 1. Absolute for DCUPD. File 2. Absolute for DCSEL. File 3. Absolute for DCRPT. File 4. Absolute for DCRET. File 5. Absolute for DCCONVT. File 6. Absolute for DCUTL. File 7. Absolute for DCIDX. File 8. Absolute for DCGEN. File 9. Absolute for DCCONGN.
*PFGDECK		DECK	DECKOPL permanent files. File 1. DECKOPL. File 2. INSTALL procedure file. File 3. COMMOD. File 4. GDIR program binaries.
PFGFSEH		FSEH/ FSES	FSE Help Files. File 1. FSEHELP file. File 2. FSTEACH file. File 3. FSEPROC file. File 4. SMFSTAT procedure.
PFGFSE1	FSE	FSE1	File 1. Contains SMF startup file.
PFGHELP		HELP	File 1. Contains HELPLIB file.
PFGIAF1	IAF	IAF1	Interactive Facility Startup procedures. File 1, record 1. Startup proc IAF. File 1, record 2. Startup proc IAFTM. File 1, record 3. Startup proc IAFTR.
PFGITF1	ITF	ITF1	File 1. Contains ITF NAMSTRT startup record.
PFGMAN1		MAN1	Optional set of online manuals: procedures, source and bound copies of the manuals. File 1. Contains installation procedures and documentation for all other files in this file.
PFGMCS1	MCS	MCS1	File 1. MCS STARTUP PROCEDURE.
*PFGMISC		MISC	File 1. Contains MISCPL.
PFGMMCL	MMCL	MMCL	MAP III/IV-2X Macro Control Library Files. File 1. MAPLIBE - MAP III, MAP IV-23/25. File 2. MAPLIBC - MAP IV-20/21.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	SYSGEN Function/Notes
PFGMAS	MAS	MAS	
PFGMSSI	MSSI	MSSI	MSSI 3 Permanent Files. File 1. MAPCMI startup procedure. File 2. MAPECS startup procedure. File 3. MAPCH startup procedure. File 4. MSSIP - misc. MSSI/MMCL procedures. File 5. MJOBSPL - MSSI test jobs PL.
PFGMSS1	MSS	MSS1	File 1. MSS startup procedure.
PFGNAMD	NAM5D	SWAP	NAM5 trace/Debug binaries. SYSGEN procedure SWAP is used to load these binaries.
PFGNAM5†	NAM5	NAM5/ NAMS	NAM startup files. File 1. NAMSTRT file. File 2. NAM startup procedure. File 3. NAMNOGO startup procedure.
PFGNDL1		NDL1	Network Definition File. File 1, record 1. NDL source file NDLDATA. File 1, record 2. If present, files 2 and 3 will also be present. File 2. Compiled NCFFILE of NDLDATA. File 3. Compiled LCFFILE of NDLDATA.
PFGNSS1	NSS	NSS1	File 1. NOS Scope Station startup procedure.
PFGONLM		ONLM	Default set of online manuals: procedures, source and bound copies of the manuals. File 1. Contains installation procedures and documentation for all other files in this file.
PFGPFT1		PFT1	Protocol File Transfer Facility. File 1. PROC/RMUGET (64-character set version) File 2. PROC/RMUGET (63-character set version) File 3. ULIB/PFTF (debug version)
*PFGPRPT		PRPT	Contains history idents of NOS PSR code in current release.

†For SYSGEN(FULL), NAMSTRT is loaded to user name NETOPS, NAM and NAMNOGO to SYSTEMX. For SYSGEN(SOURCE), NAMSTRT is loaded to user name INSTALL.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	SYSGEN Function/Notes
PFGPSUD		SWAP	Printer Support Utility binaries. SYSGEN procedure SWAP is used to load these binaries. File 1. 64-character set trace binaries. File 2. 63-character set trace binaries. File 3. 63-character set nontrace binaries.
PFGPSU1		PSU1	File 1. Contains PSU NAMSTRT startup record.
PFGSRB1		SRB1	Contains SRB in ASCII 6/12 format.
PFGRBFD	RBF5D	SWAP	RBF5 Trace/Debug binaries. SYSGEN procedure SWAP is used to load these binaries.
PFGRBF5	RBF5	RBF5	File 1. Contains RBF NAMSTRT startup record.
PFGRDF1	RDFEX	RDF1	File 1. Contains RDF startup procedure.
PFGRHP1	RHP	RHP1	File 1. Contains RHP NAMSTRT startup record.
PFGTAF1	TAF	TAF1	Transaction Facility permanent files. File 1. TAF startup procedure. File 2. TASKLIB.
PFGTLIB		TLIB	Terminal Definition Utility Files. File 1. ULIB/TERMLIB. File 2. TDUIN file. File 3. TDUFILE.
PFGXEDT	XEDIT	XEDT	File 1. Contains XEDIT help file, XEDITH.

On component orders, you also get the following files:

PRODUCT	Contains all binaries for the deadstart tape.
DSTDIR	Contains LIBEDIT directives for the products in file PRODUCT.
BLDLIB	Contains a procedure to properly update all ULIBs from binaries in file PRODUCT.

CONTROLWARE INSTALLATION D

Controlware for 834/844/885 mass storage and 639/667/669 magnetic tape devices is available on either punched cards or 9-track magnetic tapes. Cards are supplied with the standard release materials. If you want the magnetic tape versions, specify magnetic tape when ordering the controlware. The 819 Peripheral Processor Unit (PPU) driver is available only on magnetic tape.

819 PPU DRIVER INSTALLATION

The 819 PPU driver can be installed only on a model 176. The 819 PPU driver resides on the system as a PPU-type record named HCD and is loaded into the first-level peripheral processors (FLPPs) during deadstart. The installation procedure assembles the 819 driver from the release tape and copies the binaries to the permanent file PRODUCT.

To build a system with a new 819 driver, submit a job similar to the following:

```
job command.
USER,username,password,familyname.
BEGIN,HCD,INSTALL.
--eoi--
```

NOTE

This controlware can be received only if you order NOS on the OIP.

834/844/885/895 CONTROLWARE INSTALLATION

The 834/844/885/895 controlware resides on the system as a PPU-type record named BCS for half-track 7054/7152/7154 controlware, BCF for full-track 7152/7154 controlware, FMD for 7155-1 controlware, PHD for 7155-401 controlware, MA464 for full-track 7165, ISD for full track 7255-1 controlware, or CMD for control module controlware; it is loaded into the 7054, 7152, 7154, 7155-1 or 7155-401, 7165, 7255-1, or control module controller during system deadstart. To build a system with new controlware from either punched cards or magnetic tape, perform one of the following procedures. The procedures copy the 834/844/885/895 controlware program from the released deck or tape to the permanent file PRODUCT.

NOTE

The 7255-1 and control module controlware are not available in card deck form.

PUNCHED CARDS

Submit the following job to install 834/844/885/895 controlware from cards.

```
job command.
USER,username,password,familyname.
NOTE,IN.+*COMMENT PPU/typ typ FIRMWARE xxxxx-xxx(PNxxxxxxx)
COPYBF,INPUT,INHOLD.
BEGIN,typ,INSTALL.
--eor--

834/844/885/895 controlware deck

--eoi--
```

typ is the controlware type:

BCS	Half-track 7054/7152/7154.
BCF	Full-track 7152/7154.
FMD	7155-1
PHD	7155-401.
MA464	Full-track 7165.

xxxxx-xxx(PNxxxxxxxx) is the number associated with the controlware. For example, if type is FMD, the number could be MG101-A00(PN12345678).

MAGNETIC TAPE

To install 834/844/885/895 controlware from magnetic tape, submit a job similar to the following.

Job	Comments
job command.	
USER,username,password,familyname.	
LABEL,TAPE,F=SI,NT,D=1600,LB=KU.	Before entering this command, the 844/885/895 controlware tape should be mounted and ready on a 9-track tape drive. The tape must be assigned by the operator.

Job	Comments
NOTE,IN.+*COMMENT PPU/typ nn	typ is the controlware type. BCS Half-track 7054/7152/7154. CMD Control Module. ISD Full-track 7255-1. MA464 Full-track 7165. BCF Full-track 7152/7154. FMD 7155-1. PHD 7155-401. nn is the version of 844/885/895 controlware you are installing.
COPYBF,TAPE,INHOLD.	
RETURN,TAPE.	
BEGIN,typ,INSTALL.	typ is controlware type (previously defined).
--eoi--	

NOTE

The COPYBF command must be preceded by the command SKIPF,TAPE,1 if you are installing ISD controlware, and the command SKIPF,TAPE,2 if you are installing CMD controlware.

639/667/669 CONTROLWARE INSTALLATION

The 639 controlware resides on the system as a PPU-type record named CW63x; 667/669 controlware resides on the system as a PPU-type record named FIRM66X. The controlware is loaded into the 7021 or 7152 controller at system deadstart and whenever the magnetic tape subsystem, MAG, is brought to a control point. To build a system with new controlware from either punched cards or 9-track magnetic tape, perform one of the following procedures. The procedures copy the 667/669 controlware program from the released deck or tape to the permanent file PRODUCT.

NOTE

CW63x controlware installation is performed in the same manner as the 834/844/885/895 controlware. However, the COPYBF command must be preceded by a SKIPR,TAPE,1 when installing from tape.

PUNCHED CARDS

Figure D-1 illustrates the structure of the coldstart deck used to load the 639/667/669 controlware when deadstarting from 667 or 669 magnetic tape units (coldstart).

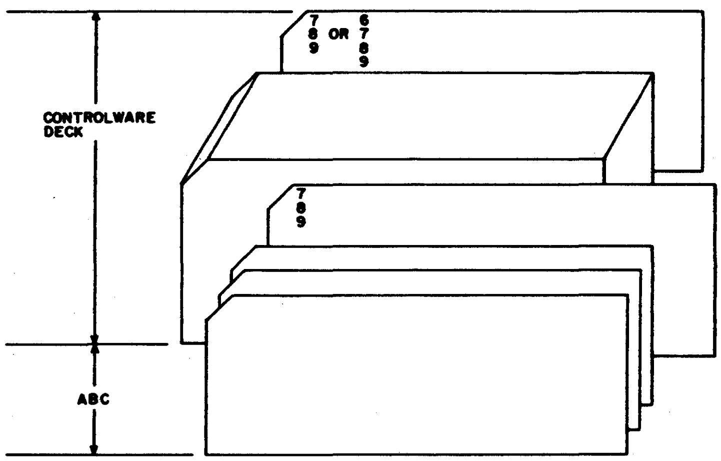

Figure D-1. Coldstart Deck Structure

There are two records in the coldstart deck, the automatic buffer controlware loader (ABC) program and the controlware deck. The program on the deadstart panel reads and executes ABC, which causes the controlware deck to be loaded. When controlware loading is complete, system or maintenance deadstart proceeds.

Use the following job to add new controlware programs to the deadstart tape and also to obtain the coldstart deck. It collects the binary card deck for the 667/669 magnetic tape controlware program FIRM66X, installs the binaries on the permanent files used to create a new deadstart tape, and punches a coldstart deck. (This coldstart deck is also supplied with the released controlware.)

Job	Comments
job command.	
USER,username,password,familyname.	
NOTE,IN.+*COMMENT PPU/FIRM66X nn	nn is the version of 639/667/699 controlware being installed.
COPYBF,INPUT,INHOLD.	
BEGIN,MTS,INSTALL.	
--eor--	
667/669 controlware deck.	
--eoi--	

MAGNETIC TAPE

To install 667/669 controlware from magnetic tape, submit a job similar to the following.

Job	Comments
job command.	
USER,username,password,familyname.	
LABEL,MTSTAPE,F=L,NT,D=1600,LB=KU.	Before entering this command, the 667/669 controlware tape should be mounted and ready on a 9-track drive. The tape must be assigned by the operator.
NOTE,IN.+*COMMENT PPU/FIRM66X nn	nn is the version of the 667/669 controlware being installed.
SKIPF,MTSTAPE.	
COPY,MTSTAPE,TEMP,TC=EOF,N=1,BS=10000B.	
RETURN,MTSTAPE.	
REWIND,TEMP.	
COPYBR,TEMP,INHOLD.	
BEGIN,MTS,INSTALL.	
RETURN,PUNCHB.	
--eoi--	

NETWORK ACCESS DEVICE (NAD) CONTROLWARE INSTALLATION

NAD controlware resides on the system as a PPU-type record named 170 for NADs connected to CYBER 170 computer systems. Controlware named IBM for NADs connected to IBM systems and MIN for NADs connected to minicomputer (DEC) systems may also be installed on the system as a PPU-type record. It is loaded into the NADs during Remote Host Facility (RHF) initialization or on operator request. To build a system with new 170 NAD controlware or to install IBM, MIN, or 200 controlware from either punched cards or magnetic tape, perform one of the following procedures. The procedures copy the NAD controlware program from the released deck or tape to the permanent file PRODUCT.

PUNCHED CARDS

Submit the following job to install NAD controlware from cards.

Job	Comments
job command.	
USER,username,password,familyname.	
NOTE,IN.+*COMMENT PPU/typ nn	typ is the controlware type. 170 CYBER 170 NADs. IBM IBM NADs. MIN Minicomputer (DEC) NADs. nn is the version of NAD controlware you are installing.

Job	Comments
COPYBF,INPUT,INHOLD.	
BEGIN,typ,INSTALL.	typ is the controlware type (previously defined).
--eor--	
NAD controlware deck.	
--eoi--	

MAGNETIC TAPE

To install NAD controlware from magnetic tape, submit a job similar to the following.

Job	Comments
job command.	
USER,username,password,familyname.	
LABEL,TAPE,F=SI,NT,D=1600,LB=KU.	Before entering this command, the NAD controlware tape should be mounted and ready on a 9-track tape drive. The tape must be assigned by the operator.
NOTE,IN.+*COMMENT PPU/typ nn	typ is the controlware type. 170 CYBER 170 NADs. IBM IBM NADs. MIN Minicomputer (DEC) NADs. nn is the version of NAD controlware you are installing.
COPYBF,TAPE,INHOLD.	
RETURN,TAPE.	
BEGIN,typ,INSTALL.	typ is controlware type (previously defined).
--eoi--	

CONTROLWARE CARD DECK GENERATION

To punch a controlware card deck, run the following job.

Job	Comments
job command. USER,username,password,familyname. COMMON,SYSTEM. GTR,SYSTEM,FILE.PPU/dname	 dname is the name of the record containing the controlware desired.
ATTACH,OPL=OPL. MODIFY,C,Z./*EDIT ABC COMPASS,I,S=NOSTEXT. COPYBR,FILE,LGO. ROUTE,LGO,DC=SB. --eoi--	If OPL is on tape, use a LABEL command.

Extract the first card from the resulting card deck before inserting the deck in subsequent jobs.

63-CHARACTER SET INSTALLATION E

If you want to install a system with the 63-character set format, you must perform the customized installation described in this appendix.

DEDICATED SYSTEM

To install a 63-character set format on a dedicated system using the released deadstart tape, follow these steps:

1. After you load the files necessary for installation (refer to section 3), execute the following command to convert the Modify-formatted DECKOPL procedures to the 63-character set format:

   ```
   BEGIN,SETUP,INSTALL,CV63,NEWPL.
   ```

2. Run the UCOMMOD procedure to recreate file COMMOD. Use the new COMMOD file to set your DECKOPL installation defaults.

3. Execute the following command to rebuild the INSTALL procedure file to include code for a 63-character set installation:

   ```
   BEGIN,SETUP,INSTALL,DF63,INSTALL,MOD=COMMOD.
   ```

   NOTE

   Any future calls to SETUP should include the DF63 parameter.

4. Run the SEED procedure.

5. Create file USER for the MODOPL procedure, if needed.

6. Execute the MODOPL procedure. The composite OPL is created in the 63-character set format.

7. Deadstart the system again using this IPRDECK entry:

   ```
   CSM=63.
   ```

8. Include the parameter CSET=C63 or CSET=A63 on the call to build TEXT.

9. Continue with the installation.

PRODUCTION ENVIRONMENT

To install a 63-character set format in a production environment system (assuming that the running system is in 63-character set format), follow these steps:

1. Before executing the MODOPL procedure, execute this command:

   BEGIN,SETUP,INSTALL,DF63,INSTALL.

2. Run SEED procedure.
3. Execute the MODOPL procedure.
4. Include the keyword CSET on the call to build TEXT.
5. Continue with the installation.

FILE CONVERSIONS FOR BOTH DEDICATED AND PRODUCTION SYSTEMS

All text files that are installed by SYSGEN should be converted to 63-character set format. Use the FCOPY command to convert the files. For example, the FSEPROC file would be converted with this command:

```
FCOPY(P=FSEPROC,PC=DIS64,N=newfile,NC=ASCII63,R)
```

The Full Screen Editor help file, FSEHELP, would be converted by this command:

```
FCOPY(P=FSEHELP,PC=ASCII64,N=newfile,NC=ASCII63,R)
```

The following files are installed by SYSGEN and must be converted to 63-character set format because they are not recreated from the build jobs.

- On user name LIBRARY, convert the ASCII 6/12 files FSEHELP, FSTEACH, TDUIN, TDUFILE, and HELPLIB. Convert the display code file FSEPROC.
- On user name INSTALL, convert display code files USERBPS, SMFSTAT, NDLDATA, PSRRPT, and ASCII 6/12 file SRB.

Because HELPLIB is a ULIB file, you must first GTR all PROCs from the file, convert them with FCOPY, and then re-LIBGEN the file.

PRINTER SUPPORT UTILITY

The Printer Support Utility (PSU) is released in binary format only. The X.SYSGEN(SWAP) procedure enables 63-character set customers to use PSU by loading a 63-character set version of PSU from the permanent file tapes. You can load a trace or nontrace version of PSU. Here's the format for the command:

```
X.SYSGEN(SWAP,dd,PSU,,,type)
```

Replace dd with the tape density of the deadstart tape that will be built; replace type with T63 to load a trace version or NT63 to load a nontrace version.

This procedure will build a new deadstart tape containing your running system plus the 63-character set PSU binaries. To use this procedure you must use SYSGEN(FULL) to load the RECLAIM database and you'll need to mount the permanent file tapes. There is no provision for restoring the original 64-character set version, so be sure to keep a copy of your original deadstart tape.

THE REMOTE DIAGNOSTIC FACILITY F

The Remote Diagnostic Facility (RDF) allows you to use an interactive terminal when your networks are not yet installed. Thus, you can use the RDF as an interface for using a NOS editor to create your deadstart deck files or configuration files.

To initiate RDF, follow these steps:

1. When you are deadstarting the system, make an EQPDECK entry that defines the two-port multiplexer to which the terminal you want to use is connected. (Check with your customer engineer (CE) to ensure that the terminal's baud rate agrees with that set on the two-port multiplexer. Refer to the NOS 2 Analysis Handbook for the syntax of the entry.)

2. After you have deadstarted the system, check the operator E,A. display for the equipment status of the two-port multiplexer. If the status is OFF, enter this command:

   ON,nn

nn	The EST ordinal of the two-port multiplexer.

3. Enable the RDF subsystem by entering these commands:

   SUBSYST.
   L.ENABLE,RDF.
   L.END.

4. Now initiate RDF by entering this command:

   RDF.

5. Press the RETURN key on the terminal a few times; the login banner is displayed at the terminal.

Using RDF is similar to using IAF/NAM, with the following exceptions:

- To correct input, press the backspace key or CTRL-H.
- To delete input, press the ESCAPE key.
- To interrupt execution, press I or the BREAK key.
- To terminate execution, press S or enter STOP.
- To exit TEXT mode, press CTRL-C or the BREAK key.
- The typing-ahead feature is not available on RDF.
- Full Screen Editor (FSE) does not function properly.

SYSTEM CONFIGURATIONS G

To minimize installation time and confusion and to establish conventions which will enable better CDC service/support to users, the preferred configurations have been defined in table G-1.

Complete configurations are not defined in table G-1 due to the differing customer requirements. Conventions are specified for a base system. Systems configured according to the conventions will provide the following benefits:

- Simplify the selection of peripheral equipment channels by the customer and customer engineering.

- Allow the customer to easily deadstart NOS using the pre-defined configurations released on the deadstart tape.

Table G-1. Preferred Configuration

Channel	CYBER 170/180-800 Models 815, 825, 835, 840, 845 850, 855, 860, 865, 875	CYBER 180 Models 810, 830		
		a	b	c
0	+	RMS-ISD_1	RMS-ISD_1	RMS-ISD_1
1	ROTATING MASS STORAGE $(RMS)_1$	+	RMS_1	RMS-ISD_3
2	RMS_2	–	–	–
3	RMS_3	+	RMS_2	RMS-ISD_4
4	RMS_4	+	RMS_3	+
5	$COMMUNICATIONS_1$	–	–	–
6	$COMMUNICATIONS_2$	MT-$ISMT_1$	MT-$ISMT_1$	MT-$ISMT_1$
7	$COMMUNICATIONS_3$	COMMUNICATIONS	$COMMUNICATIONS_1$	COMMUNICATIONS
10	CONSOLE	CONSOLE*	CONSOLE*	CONSOLE*
11	MAG $TAPE_1$	+	+	+
12	UNIT RECORD	+	UNIT RECORD	+
13	+	–	–	–
20	+	RMS-ISD_2	RMS-ISD_2	RMS-ISD_2
21	RMS_5	+	RMS_4	+
22	RMS_6	–	–	–
23	RMS_7	+	+	+
24	RMS_8	+	+	+
25	+	–	$COMMUNICATIONS_2$	–
26	+	ISD/ISMT	ISD/ISMT	ISD/ISMT
27	+	+	+	+
30	+	–	MAG $TAPE_1$	–
31	MAG $TAPE_3$	+	MAG $TAPE_2$	+
32	MAG $TAPE_4$	+	+	+
33	MAG $TAPE_2$	–	–	–

a: Specifies channel assignments for a 810/830 using only Intelligent Small Disk (ISD) and Intelligent Small Tape (ISMT) disk and tape peripherals.

b: Specifies channel assignments for a 810/830 with both ISD/ISMT peripherals and other CDC disk and tape peripherals.

c: Specifies channel assignments for the C180-810 and 830 with the 18102-2 channel option.

+: Indicates that the channel is available for CYBER channel peripheral connection.

–: Indicates that the channel is not available CYBER channel for peripheral connection.

*: 18002-1 console only.

INDEX

COMMENT SHEET

MANUAL TITLE: CDC NOS Version 2 Installation Handbook

PUBLICATION NO.: 60459320 REVISION: E

NAME:

COMPANY:

STREET ADDRESS:

CITY: STATE: ZIP CODE:

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

☐ Please Reply ☐ No Reply Necessary

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

CUT ALONG LINE

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION

Publications and Graphics Division
ARH219
4201 North Lexington Avenue
Saint Paul, Minnesota 55112

CUT ALONG LINE

FOLD FOLD

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION