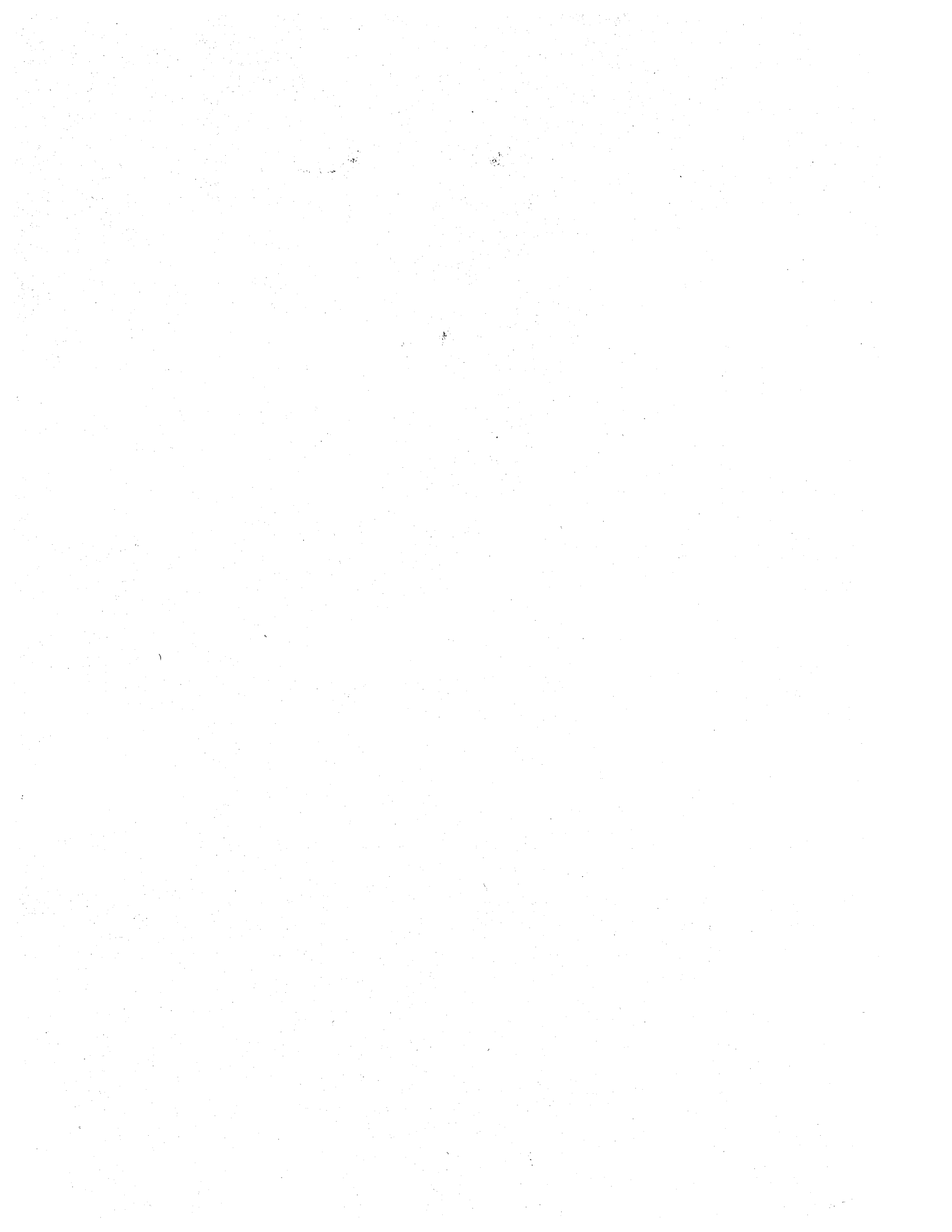


 ***NOS Version 2***
Installation Handbook



NOS Version 2

Installation Handbook

Installation

NOS Version 2

Installation Handbook

Installation

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features and parameters.

Manual History

Revision	System Version	PSR Level	Date
A	2.0	562/552	April 1982
B	2.1	580/577	January 1983
C	2.2	596/587	October 1983
D	2.3	617	October 1984
E	2.4.1	630	March 1985
F	2.4.2	642	September 1985
G	2.4.3	647	December 1985
H	2.5.1	664/650	September 1986
J	2.5.2	678/670	April 1987
K	2.6.1	700/688	April 1988
L	2.7.1	716	December 1988
M	2.7.1	739	December 1989
N	2.7.2	774	June 1991
P	2.7.4	797	June 1992
R	2.8.3	840	August 1994

Revision R of this manual, printed August 1994, reflects NOS 2.8.3 at PSR level 840.

This edition obsoletes all previous editions.

Contents

About This Manual	7
Purpose	7
Audience	7
Manual Organization	8
Conventions	9
Submitting Comments	10
Customer Support Hotline	10
Related Publications	11
Introduction	1-1
Installation Process	1-1
Types of Installations	1-1
Customizing Installations	1-1
Dual State Installations	1-2
Preparing for an Installation	1-3
First-Time Installation	2-1
Introduction	2-1
Step 1: Prepare for the Installation	2-2
Step 2: Deadstart the System	2-9
Step 3: Install Permanent Files	2-11
Step 4: Bring Up a Network Terminal	2-12
Step 5: Configure the System	2-14
Step 6: Deadstart the New System	2-15
Step 7: Wrap Up	2-16
Upgrade Installation	3-1
Introduction	3-1
Step 1: Create New User Names	3-2
Step 2: Set Up Installation Tools	3-3
Step 3: Update Decks and Configuration Files	3-4
Step 4: Write a New Deadstart Tape	3-6
Step 5: Override Automatic File Installation	3-7
Step 6: Install Permanent Files	3-12
Step 7: Deadstart the New System	3-13
Component Order Installation	4-1
Introduction	4-1
Step 1: Create New User Names	4-2
Step 2: Set Up Installation Tools	4-2
Step 3: Update Configuration Files and Decks	4-6
Step 4: Write a New Deadstart Tape	4-7
Step 5: Override Automatic File Installation	4-8
Step 6: Install Permanent Files	4-10
Step 7: Deadstart the	4-11

Corrective Code Installation	5-1
SOLVER Modification Sets	5-1
CDCNET Batch Critical Update (BCU)	5-5
Dual State Batch Critical Update (BCU)	5-8
Customizing Installations	6-1
Introduction	6-1
Customizing Deadstart Tape Binaries	6-2
Customizing Permanent Files	6-23
Special Product Installation Information	7-1
Subsystem Initiation	7-1
AAM2 - CYBER Record Manager Advanced Access Methods Version 2	7-4
APL2 - APL Version 2	7-7
ATF - Automated Tape Facility	7-9
BAM - CYBER Record Manager Basic Access Methods Version 1	7-10
BASIC3 - Basic Version 3	7-11
CCL - CYBER Control Language Version 1	7-12
CCP - CYBER CROSS System and Communications Control Program	7-15
CDCNET - Control Data Distributed Communications Network	7-55
CDCS2 - CYBER Database Control System Version 2	7-77
CID - CYBER Interactive Debug Version 1	7-80
CML - Concurrent Maintenance Library	7-80
COBOL5 and COBOL5Q - COBOL Version 5	7-81
DCL - Data Catalogue 2 Version 2.0	7-83
DUAL - Dual State	7-84
FDBF - FORTRAN Data Base Facility Version 1	7-99
FSE - Full Screen Editor	7-100
FTN4, FTNTS, FTN5 (FORTRAN Extended Version 4, FORTRAN Extended Version 4 with Interactive Option, FORTRAN 5)	7-104
IAF - Interactive Facility Version 1	7-105
ITF - Interactive Transfer Facility Version 1	7-107
LCS3 and FCS3 - Conversion Aids System Version 3	7-109
LOADER - CYBER LOADER Version 1	7-111
MAP Subsystem	7-113
MCS - Message Control System Version 1	7-115
MSE - Mass Storage Extended Subsystem	7-118
NAM5/NAM5D - Network Access Method Version 1	7-120
NIP5870 - 5870 Printer	7-142
NOS and NOS2B - Network Operating System	7-143
NSS - NOS Scope 2 Station Facility 1	7-164
PSU - Printer Support Utility Version 1.1	7-171
QU3 - Query Update Version 3	7-174
RBF5/RBF5D - Remote Batch Facility Version 1	7-175
RDFEX - Remote Diagnostic Facility	7-177
RHF - Remote Host Facility Version 1	7-178
RHP - PTF/QTF File Transfer Facility	7-182
SORT5 - Sort/Merge Version 5	7-186
TAF - Transaction Facility Version 1	7-187
TCP/IP/FTP/TELNET	7-198
TERMLIB - Terminal Library	7-204
TEXT and TEXTIO - Product TEXTS and Product TEXTS I/O	7-205

SYSGEN **8-1**

 Introduction 8-1

 Calling SYSGEN 8-1

 Loading Permanent Files 8-2

 SYSGEN Functions 8-5

 SYSGEN Maintenance 8-9

 SYSGEN Validations 8-10

Installation Commands, Parameters, and Procedures **9-1**

 Introduction 9-1

 COMMOD File Parameters 9-1

 DECKLIS Procedure 9-9

 Disk Installations 9-10

 GENDST Procedure 9-14

 MISCGET Procedure 9-15

 REPORT Procedure 9-15

 RESETP Procedure 9-16

 SETUP Procedure 9-17

Glossary **A-1**

File Formats **B-1**

 Order of Products B-1

 Permanent File Tapes B-3

63-Character Set Installation **C-1**

 Installation Procedure C-1

 File Conversions C-2

The Remote Diagnostic Facility **D-1**

System Configurations **E-1**

Index **Index-1**

Figures

6-1. Build Dependencies Chart (Group 1) 6-13

6-2. Build Dependencies Chart (Groups 2 and 3) 6-14

6-3. Build Dependencies Chart (Groups 4, 5, and 6) 6-15

7-1. CCP/CROSS Build Step Dependencies 7-20

7-2. CCP File Dependencies 7-21

7-3. Integration of Program Libraries in CCP Installation Process 7-22

7-4. Network Configuration - Example 1 7-44

7-5.	USERBPS Definitions, NDL Source Input, and EQPDECK Entries for Example 1 (Continued)	7-46
7-6.	Network Configuration - Example 2	7-49
7-7.	USERBPS Definitions, NDL Source Input and EQPDECK Entries for Example 2 (Continued)	7-50
7-8.	Example of Parameter Record	7-128
7-9.	Example of Job Skeleton Record (Continued)	7-130

Tables

2-1.	Software Products and Deadstart Deck Requirements	2-4
2-2.	Software Products and Configuration Files Required	2-4
2-3.	Preconfigured Deadstart Decks (NOS 810/830 Decks)	2-6
2-4.	Preconfigured Deadstart Decks (NOS Non-810/830 Decks)	2-7
2-5.	Preconfigured Deadstart Decks (Dual State Decks)	2-8
3-1.	Automatic File Installation	3-9
6-1.	OIP Options and Associated Installation Procedures	6-10
6-2.	Group 1 Products	6-16
6-3.	Group 2 Products	6-18
6-4.	Group 3 Products	6-19
6-5.	Group 4 Products	6-19
6-6.	Group 5 Products	6-20
6-7.	Group 6 Products	6-21
7-1.	Subsystem Initiation	7-2
7-2.	Format of First Three Words of Compression/Decompression Routines	7-4
7-3.	Recommended Limits for APL0 and APL1	7-8
7-4.	Options and Variants with CCP	7-17
7-5.	Released CCP Variants	7-18
7-6.	CCP/CROSS Tape and Disk File Requirements	7-24
7-7.	Table Sizes and Central Memory Requirements	7-109
7-8.	NOS Common Decks	7-145
7-9.	Released Values of Permanent File Limit Ranges	7-156
7-10.	SPOT Octal Memory Requirements	7-167
7-11.	Binaries Built	7-184
8-1.	SYSGEN User Names and Passwords	8-11
9-1.	Source File and Auxiliary PL Assignments	9-12
E-1.	Preferred Configuration	E-2

About This Manual

Purpose

This handbook describes the installation of the CONTROL DATA® Network Operating System (NOS) Version 2.8.3 Level 840 and its products. NOS controls the operation of the following computer systems:

- CONTROL DATA CYBER 180 Computer Systems
Models 810, 830, 835, 840, 845, 850, 855, 860, 870, 960, 970, 990, 994, and 995
- CONTROL DATA CYBER 170 Computer Systems
Models 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 815, 825, 835, 845, 855, 865, and 875
- CONTROL DATA CYBER 70 Computer Systems
Models 71, 72, 73, and 74
- CONTROL DATA 6000 Computer Systems

Audience

Users who do not intend to modify the software 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 Version 2 Operations Handbook and the NOS Version 2 Reference Set, Volume 3.

Users who intend to customize the software during installation should be familiar with NOS, COMPASS, and each product to be customized. In other words, you should be familiar with the information provided in the NOS Version 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:

- Chapter 1 summarizes the four types of NOS installations, discusses dual state installations, and gives general guidelines for beginning all types of installations.
- Chapter 2 describes how to install a full order of NOS and its product set for the first time on a dedicated system.
- Chapter 3 describes how to install a new version of NOS on a system currently running a prior version of NOS.
- Chapter 4 describes how to install additional products on a system currently running NOS.
- Chapter 5 describes how to install corrective code and Batch Critical Update (BCU) code to a system currently running NOS.
- Chapter 6 describes how to customize deadstart tape binaries and permanent files.
- Chapter 7 describes subsystem initiation, and provides special information for configuring and customizing NOS and each of its products.
- Chapter 8 describes the SYSGEN utility.
- Chapter 9 describes special installation commands, parameters, and procedures.
- The five appendixes contain: a glossary of terms, the file formats of the source materials, special information for a 63-character set installation, Remote Diagnostic Facility information, and system configurations.

Conventions

The following conventions are used throughout this handbook:

- Technical changes are indicated by bars in the margins.
- The values `insun`, `netopun`, and the `netadun` should be substituted throughout this handbook with the user names that the site is using for the installation user name, network operations user name, and network administration user name, respectively. The Control Data default values are `INSTALL`, `NETOPS`, and `NETADMN`.
- 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, 74, and 76 only.
- The terms `command` and `control statement` are interchangeable.
- The forms of extended memory are described as follows:
 - Extended memory for models 865 and 875 and CYBER 180-class machines is unified extended memory (UEM) and may also include either extended core storage (ECS), extended semiconductor memory (ESM), or STORNET.
 - Extended memory for model 176 is large central memory extended (LCME) and may also include ECS or ESM.
 - Extended memory for all other NOS computer systems is either ECS, ESM, or STORNET.¹
- ECS refers to ECS, ESM, and STORNET and extended memory refers to all forms of extended memory unless otherwise noted. However, when referencing extended memory in the context of an ECS multiframe complex, a distributive data path (DDP), or a low-speed port (LSP) access, UEM and LCME are excluded. ECS, ESM, and STORNET are the only forms of extended memory that can be shared in an ECS multiframe complex and can be accessed
- Uppercase letters within command formats must be entered exactly as given; replace lowercase letters with appropriate characters as described in the text.
- The values `psrin` and `psrout` are substituted throughout the text for actual PSR numerical values. The value `psrin` refers to the NOS release level and `psrout` refers to the PSR level of any products you customize when building products from source.
- At all sites, you must specify `Job` and `USER` commands. The `CHARGE` command is not required at all sites.
- The CONTROL DATA 18002-2 console is available as an option for certain CYBER 180-class machines. This product includes a 721-21 display terminal and an AV117A cable. This console is referred to throughout this handbook as CC634B.
- The CONTROL DATA 19003-3 console is available as an option for certain CYBER 180-class machines. This product includes a video monitor, keyboard, 40-Mbyte hard

1. STORNET is only supported on CYBER 170 (except model 176) and 180 Computer Systems.

disk (Winchester) drive, 1.2-Mbyte 5-1/2-in. floppy disk drive, 640-Kbyte RAM memory, one parallel printer port, and nine RS-232-C serial ports. This console is referred to throughout this handbook as CC598B.

- Press CR means press the carriage return key on the CC545 console, press the Enter/Return key on the CC598B console, or press the NEXT key on the CC634B console.

Submitting Comments

If you have comments concerning this manual, mail your comments to:

Control Data
Information Services ARH219
4201 Lexington Avenue N.
St. Paul, MN 55126-6198

If you have access to SOLVER, an online facility for reporting problems, you can use it to submit comments about this manual. When entering your comments, use NS2 as the product identifier. Include the name and publication number of the manual.

If you have questions about the packaging and/or distribution of printed manuals, call or write Literature and Distribution Services as described in Related Publications, later in this preface.

Customer Support Hotline

Control Data's Central Software Support maintains a hotline to assist you if you have difficulty using our products. If you need help not provided in the documentation, or find that the product does not perform as described, call one of the following numbers. A support analyst will work with you.

From the USA and Canada: 1-800-345-6628

From other countries: 1-612-482-3434

Related Publications

To order a Control Data manual, send an order form to:

Control Data
Literature and Distribution Services, ARHLDS
4201 Lexington Avenue N.
St. Paul, MN 55126-6198

To obtain an order form or to get more information, call (612)482-3800 or (612)482-3801, or FAX your enquiry to (612)482-3813. (If you are a Control Data employee, use the Controlnet number 235-3800, 235-3801, or 235-3813.)

The NOS System Information Manual is an online manual that includes brief descriptions of all NOS and NOS product manuals. To access this manual, log in to NOS and enter the command EXPLAIN.

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

CDCNET Manuals

Control Data Publication	Publication Number
CDCNET Configuration Guide	60461550
CDCNET Network Operations and Analysis	60461520
CDCNET TCP/IP Programming Interfaces and Applications	60000214

Extended Memory Manuals

Control Data Publication	Publication Number
CYBER 5380-100 STORNET Subsystem (SNSS) Hardware Reference Manual	60000188
Extended Core Storage Reference Manual	60347100
Extended Core Storage II and Distributive Data Path Reference Manual	60430000
Extended Semiconductor Memory Hardware Reference Manual	60455990

Hardware Manuals

Control Data Publication	Publication Number
CYBER 70 Model 71 Computer System Hardware Reference Manual	60453300
CYBER 70 Model 72 Computer System Hardware Reference Manual	60347000
CYBER 170 Computer Systems Models 171 through 175 (Levels A, B, C) Model 176 (Level A, B, C) Hardware Reference Manual	60420000
CYBER 170 Computer Systems Models 720, 730, 740, 750, and 760 Model 176 (Level B/C) Hardware Reference Manual	60456100
CYBER 170 Computer Systems Models 815 and 825 Hardware Reference Manual	60469350
CYBER 170 Computer Systems Models 835, 845, and 855 CYBER 180 Computer Systems Models 835, 840, 845, 850, 855, 860, 970, and 990 CYBER 990E, 995E, and 994 Computer Systems CYBER 170 State Hardware Reference Manual	60469290
CYBER 170 Computer Systems Models 835, 845, and 855 CYBER 180 Computer Systems Models 835, 845, and 855 Hardware Operator's Guide	60458390
CYBER 170 Computer Systems Models 865 and 875 Hardware Reference Manual	60458920
CYBER 180 Models 810 and 830 Computer Systems Hardware Operator's Guide	60469440
CYBER 180 Models 810 and 830 Computer Systems Hardware Reference Manual	60469420
CYBER 840A, 850A, 860A, and 870A Computer Systems Hardware Reference Manual	60463560
CYBER 960 Computer Systems CYBER 170 State Hardware Reference Manual	60000127
19003-3 System Console CC598B Hardware Operation/ Maintenance Guide	60463610
380-170 Network Access Device Hardware Reference Manual	60458500
5830 Disk Array Subsystem Configuration Guide	60000494

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) Reference Manual	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 1.5 Reference Manual	60495700
FORM Version 1 Reference Manual	60496200
Modify Version 1 Reference Manual	60450100
NOS Online Maintenance Software Reference Manual	60454200
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 Full Screen Editor User's Guide	60460420
NOS Version 2 Operations Handbook	60459310
NOS Version 2 Reference Set, Volume 3, System Commands	60459680
NOS Version 2 Reference Set, Volume 4, Program Interface	60459690
NOS Version 2 Screen Formatting Reference Manual	60460430
NOS Version 2 Security Administrator's Handbook	60460410
NOS Version 2 Systems Programmer's Instant	60459370
SYMPL Version 1 Reference Manual	60496400

Optional Product Manuals

Control Data Publication	Publication Number
APL Version 2 Reference Manual	60454000
BASIC Version 3 Reference Manual	19983900
COBOL Version 5 Reference Manual	60497100
Concurrent Maintenance Library Reference Manual	60455980
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	60481400
CYBER Interactive Debug Version 1 Reference Manual	60481400
Data Catalogue 2 Version 2 Reference Manual	60483350
DMS-170 CYBER Database Control System Version 2 Application Programming Reference Manual	60485300
DMS-170 CYBER Database Control System Version 2 Data Administrator Reference Manual	60485200
DMS-170 Query Update/CYBER Record Manager Data Administration Reference Manual	60482100
FORTTRAN Database Facility Version 1 Reference Manual	60482200
FORTTRAN Extended Version 4 Common Library Mathematical Routines Reference Manual	60498200
FORTTRAN Extended Version 4 Reference Manual	60497800
FORTTRAN Extended Version 4 to FORTRAN Version 5 Conversion Aids Program Version 1 Reference Manual	60483000
FORTTRAN Version 5 Common Library Mathematical Routines Reference Manual	60483100
FORTTRAN Version 5 Reference Manual	60481300
Message Control System Version 1 Reference Manual	60480300
MSSI Version 3 Installation Handbook	60458830
MSSI Version 3 Reference Manual	60458820

Control Data Publication	Publication Number
Network Access Method Version 1 Host Application Programming Reference Manual	60499500
Network Access Method Version 1 Network Definition Language Reference Manual	60480000
Network Access Method Version 1/Communications Control Program Version 3 Terminal Interfaces Reference Manual	60480600
Pascal Version 1.1 Reference Manual	60497700
Query Update Version 3 Reference Manual	60498300
Remote Batch Facility Version 1 Reference Manual	60499600
Remote Host Facility Access Method Reference Manual	60459990
Sort/Merge Version 5 Reference Manual	60497500
TAF Version 1 Reference Manual	60459500
TAF/CRM Data Manager Version 1 Reference Manual	60459510
Update Version 1 Reference Manual	60449900
XEDIT Version 3 Reference Manual	60455730
8-Bit Subroutines Version 1 Reference Manual	60495500

Introduction

Installation Process	1-1
Types of Installations	1-1
Customizing Installations	1-1
Dual State Installations	1-2
Preparing for an Installation	1-3
Collect All Release Tapes	1-3
Collect and Update Manuals	1-4
Read All Release Bulletins	1-4
Establish Job Environment	1-4

Installation Process

Installation is the process of creating an operating system deadstart file and then using the deadstart file to load the operating system and its products into the computer. 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, including configuration files that describe the hardware and software components of your system. Most of the permanent files you need are released on the permanent file tapes. You create the configuration files for your system using sample configuration files provided by Control Data.

Types of Installations

Four types of NOS system installations are described in this handbook:

- **First-Time Installation.** Chapter 2 describes how to install a full order of NOS and its product set on a dedicated system. This type of installation is started from the system console. Instructions are included for bringing up an interactive terminal to aid in the installation process.
- **Upgrade Installation.** Chapter 3 describes how to install a new version of NOS on a system currently running a prior version of NOS. Many of the activities performed in this installation can be initiated from an interactive terminal.
- **Component Order Installation.** Chapter 4 describes how to install additional products on a system currently running NOS. The PSR level of the components you want to install must be the same as that of the NOS software running on your system.
- **Corrective Code Installation.** Chapter 5 describes how to install corrective code and Batch Critical Updates (BCUs) to a system currently running NOS.

Customizing Installations

Regardless of the type of installation you perform, you can customize NOS or other products during installation. If you want to customize portions of the software you install, first follow the steps for the type of installation you will be performing (described in chapter 2, 3, 4, or 5). When you are directed to do so, refer to chapters 6 and 7 for specific information about customizing the installation procedures and specific products.

Dual State Installations

If you want to install a dual state system (that is, one that concurrently runs both NOS and NOS/VE), refer to the chart below for instructions on when to refer to this installation handbook and when to refer to the NOS/VE Installation Handbook.

If you want to	Then
Perform a first-time installation of NOS and NOS/VE	<ol style="list-style-type: none"> 1. Follow the instructions in this handbook for a first-time NOS installation. 2. Once you have completed the NOS installation (including dual state preparations) and have re-deadstarted your system, go to the NOS/VE Installation Handbook and follow the instructions for an initial installation.
Add dual state and upgrade your NOS system at the same time	<ol style="list-style-type: none"> 1. Follow the instructions for an upgrade installation in this handbook. 2. Once you have completed the NOS installation (including dual state preparations) and have re-deadstarted your system, go to the NOS/VE Installation Handbook and follow the instructions for an initial installation.
Add dual state and <i>not</i> update your current NOS system	<ol style="list-style-type: none"> 1. Follow the instructions for an upgrade installation in this handbook. However, skip the steps that describe how to update NOS products. 2. Once you have completed dual state preparations and have re-deadstarted your system, go to the NOS/VE Installation Handbook and follow the instructions for an initial installation.
Upgrade dual state and NOS	<ol style="list-style-type: none"> 1. Follow the instructions in this handbook for an upgrade installation. 2. Once you have completed the NOS installation and have re-deadstarted your system, go to the NOS/VE Installation Handbook and follow the instructions for an upgrade installation.

If you want to	Then
Upgrade only dual state and <i>not</i> upgrade NOS	<ol style="list-style-type: none"> 1. Follow the instructions specified in Dual State on Older NOS Systems in the DUAL section of chapter 7. 2. Go to the NOS/VE Installation Handbook and follow the instructions for an upgrade installation.
Upgrade only NOS and <i>not</i> upgrade dual state	<ol style="list-style-type: none"> 1. Follow the instructions in this handbook for an upgrade installation. Note that you will be instructed to reassemble the dual state binaries.
Apply corrective code to a dual state system	<ol style="list-style-type: none"> 1. Go to chapter 5 of this handbook for information about applying a BCU to a dual state system. For information about applying corrections to NOS/VE, refer to the NOS/VE Installation Handbook.

Preparing for an Installation

Before you begin any type of NOS installation, you need to:

- Collect all release tapes
- Collect and update manuals
- Read all release bulletins

These tasks are described in the following paragraphs.

Collect All Release Tapes

You need the following 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 Order Information Package (OIP).
- Permanent File Tapes. These 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 (not NOS) or received a Batch Critical Update (BCU), the executable programs and source code for your component order are released on permanent file tapes and you will not receive a deadstart tape.

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.

Read All Release Bulletins

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

A Software Release Bulletin (SRB) is issued with each NOS release and 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 copy of the SRB is released on a separate tape.

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 Control Data.
- Check the status of problem reports.
- Search for similar problems reported by other customers.

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

Establish Job Environment

Before you begin the installation, you should remove the system prolog, user prolog, and SHELL program name from the validation file for the installation user name. To accomplish this, do the following:

- Disable SHELL command processing with the installation user name (CDSHELL, RMSHELL).
- Ensure the installation jobs have the job control software register *R1G* set to zero prior to the first installation-related command in the job; for example, SYSGEN.

Introduction	2-1
Step 1: Prepare for the Installation	2-2
Hardware Inventory	2-2
Software Inventory	2-3
Deadstart Deck Requirements	2-3
Configuration File Requirements	2-4
Initial Deadstart Preparation	2-5
Step 2: Deadstart the System	2-9
Step 3: Install Permanent Files	2-11
Step 4: Bring Up a Network Terminal	2-12
NAM/CDCNET (DI Requirements and Information)	2-12
NAM/CCP (255x Requirements and Information)	2-13
Activating the Network	2-13
Step 5: Configure the System	2-14
Deadstart Decks	2-14
Configuration Files	2-14
Step 6: Deadstart the New System	2-15
Step 7: Wrap Up	2-16
User Names and Passwords	2-16
Installation of Online Manuals	2-16
Installation of Concurrent Maintenance Library (CML)	2-16

Introduction

This chapter describes how to install a full order of NOS and its product set for the first time on a dedicated system. This type of installation is started from the system console and includes instructions for bringing up an interactive terminal to aid in the installation process.

A first-time installation consists of the steps listed below. These steps are described in the remaining sections of this chapter:

1. Prepare for the Installation
2. Deadstart the System
3. Install Permanent Files
4. Bring Up a Network Terminal
5. Configure the System
6. Deadstart the New System
7. Wrap Up

If you want to customize any deadstart tape binaries or permanent files, first complete the first-time installation instructions in this chapter and then continue with your installation by following the directions in chapter 6 of this handbook.

If you are installing a dual state system, install NOS according to the steps in this chapter. Then follow the dual state instructions in the NOS/VE Installation Handbook.

Step 1: Prepare for the Installation

Your computer system consists of a unique combination of hardware and software components. During the installation process, you must define the attributes of your mainframe, the hardware devices connected to your mainframe, how the hardware should be used by the system, the software installed at your site, and how the software should be used by the system.

The process of defining the hardware and software components of your system is called configuring the system. Configuring the system consists of two steps: determining your site's hardware and software components; and then, based on these components, creating deadstart decks and configuration files to store the information.

This section gives guidelines for gathering the configuration information you need to create deadstart decks and configuration files. This section covers the following topics:

- Hardware Inventory
- Software Inventory
- Deadstart Deck Requirements
- Configuration File Requirements
- Initial Deadstart Preparation

Hardware Inventory

Get a description of your site's computer system hardware configuration from your hardware installer. This description should include the attributes of your mainframe and the peripherals connected to the computer.

For the mainframe, determine:

- Amount of physical memory (how many megabytes)
- Model number (for example, 180-860)
- Number of central processing units (CPUs)
- Number of peripheral processors (PPs)
- Number and types of channels

For each peripheral device, determine:

- Peripheral type (for example, 895 disk)
- Channel numbers to which each peripheral is connected
- Peripheral equipment number and unit number
- Other information unique to the type of peripheral

If your site contains multiple mainframes, you should also determine what other mainframes may share or use these peripherals.

Once you have gathered all the hardware information for your site, use this information to create a configuration chart showing the layout of your hardware. This chart will assist you when you create deadstart decks and configuration files.

Software Inventory

To determine the software components of your system, refer to the Component List report which is included with your release tapes. This report, produced by Software Manufacturing and Distribution (SMD), lists the names and product numbers of all software shipped with your order. In addition, it lists the software options selected for each product. This information is based on the software options selected on the Order Information Package (OIP) for your site. You should compare the Component List report with the OIP to verify that you have received the products that were ordered.

Deadstart Deck Requirements

Configuration information is stored in text records on the deadstart tape called deadstart decks. There are five types of deadstart decks:

- **CMRDECK** (Central Memory Resident Deck). CMRDECK entries describe how central memory is used during system operation.
- **EQPDECK** (Equipment Deck). EQPDECK entries describe how hardware components (such as disks, tapes, printers, network hardware, and other equipment) are connected to the mainframe and how these components are used by the system.
- **APRDECK** (Auxiliary Mass Storage Parameter Deck). APRDECK entries identify locations on mass storage (disks) that are unusable or flawed.
- **IPRDECK** (Installation Parameter Deck). IPRDECK entries specify the default mode of system operation.
- **LIBDECK** (Library Edit Deck). LIBDECK entries specify the attributes of the programs on the deadstart tape.

Refer to the NOS Version 2 Analysis Handbook for more information about deadstart deck entries for both hardware and software. Refer to chapter 7 in this handbook for deadstart deck information for specific software products.

The released deadstart tape contains preconfigured deadstart decks that describe several standard mainframe configurations. Use these preconfigured decks as a starting point to create deadstart decks for your site. You can also use the preconfigured decks for the first deadstart of the system before installing and configuring the system. Tables 2-3, 2-4, and 2-5 list the released deadstart decks. Refer to Initial Deadstart Preparation later in this step for information about using the decks during deadstart.

In general, each hardware component requires one or more entries in the EQPDECK. The software products that require deadstart deck entries are listed in table 2-1.

Table 2-1. Software Products and Deadstart Deck Requirements

Product	Deadstart Decks
ATF	EQPDECK
Dual State	CMRDECK, EQPDECK
Mass Storage Extended	EQPDECK
NAM/CCP Network	EQPDECK
NAM/CDCNET Network	EQPDECK
NOS	CMRDECK, EQPDECK, APRDECK
NOS Scope Station Facility	CMRDECK, EQPDECK
PTF/QTF File Transfer Facility	CMRDECK
Remote Host Facility	CMRDECK, EQPDECK

Configuration File Requirements

Some software products require configuration files. To install these products, you must create the required configuration files. Table 2-2 lists the software products that require configuration files and the corresponding file names. Refer to chapter 7 of this handbook for specific configuration information for each of the products listed in this table.

Table 2-2. Software Products and Configuration Files Required

Product	Configuration Files
Dual State	LIDCMid, LIDVEid
Mass Storage Extended	MSECONF
NAM/CCP Network	LCFFILE, NCFFILE, NLFFILE
NAM/CDCNET Network	CDCNET ¹ , LCFFILE
NOS Scope Station Facility	LIDCMid
Printer Support Utility	CDCNET ¹ , EVFULFN, LCFFILE, NCFFILE
PTF/QTF (NAM/CCP)	LCFFILE, LIDCMid, NCFFILE, NLFFILE
PTF/QTF (NAM/CDCNET)	CDCNET ¹ , LCFFILE, LIDCMid
PTF/QTF (RHF)	LIDCMid, RCFMid
Remote Host Facility	LIDCMid, RCFMid
TCP/IP/FTP/TELNET	TCPHOST

1. CDCNET has numerous configuration files.

Initial Deadstart Preparation

The first time you deadstart the system (refer to Step 2: Deadstart the System), use the preconfigured deadstart decks released on the deadstart tape. The preconfigured decks are listed in tables 2-3, 2-4, and 2-5.

There are three types of preconfigured decks:

- CYBER 810/830 mainframes running NOS.
- Other mainframes running NOS.
- Mainframes running dual state systems.

All the decks include a 255x NPU configured on channel 7, node 1 and a CDCNET MDI/MTI configured on channel 7. Table sizes (CMRDECK) are set according to memory size. Main memory and ESM sizes are given in 60-bit words, except for dual state decks (42-44), which use megabytes.

- CMRD00 is a directory for all released deadstart decks.
- CMRD35 is an unconfigured CMRDECK, therefore EQPD35 contains no equipment definitions.
- IPRD01 is for the 810/830, which uses the asynchronous printer and 639 tape drives.
- IPRD00 is for all other mainframes.
- LIBD00 is for mainframes without UEM or ESM.
- LIBD01 is for mainframes with ESM.
- LIBD02 is for mainframes with UEM.

Select the decks that describe a configuration that closely resembles your site's configuration. You should not need to modify the default CMRDECK entries in the released deck. However, you will need to make EQPDECK entries to define the physical locations of a few system and temporary file disks, all the disks in your default family, your tape drives, and your printer. You should also properly define any UEM and the channel and equipment numbers for your 255x NPU or CDCNET DI. For this initial deadstart, you do not need to define any disk or tape equipment that will be assigned to NOS/VE in your actual production system. Also, you do not need to define equipment for RHF, MSE, or NOS Scope Station.

Table 2-3. Preconfigured Deadstart Decks (NOS 810/830 Decks)

Deck Number	Memory Size	Equipment Type	Channel	UN or EQ
01	262K	639 Tape	CH = 6	UN = 0
		836 Disk	CH = 0	
		ASY Printer ¹		
02	262K	639 Tape	CH = 6	UN = 0,1
		836 Disk	CH = 0	
		ASY Printer ¹		
36	262K	639 Tape	CH = 6	UN = 0
		834 Disk	CH = 0	
		ASY Printer ¹		
37	262K	639 Tape	CH = 6	UN = 0,1
		834 Disk	CH = 0	
		ASY Printer ¹		
03	512K	639 Tape	CH = 6	UN = 0,1
		836 Disk	CH = 0	
		ASY Printer ¹		
04	512K	679 Tape	CH = 31	UN = 40,41 EQ = 5
		885 Disk	CH = 1	
		580 Printer	CH = 12	
40	512K	639 Tape	CH = 6	UN = 0,1
		834 Disk	CH = 0	
		ASY Printer ¹		
05	≥1000K	639 Tape	CH = 6	UN = 0,1
		836 Disk	CH = 0	
		ASY Printer ¹		
06	≥1000K	679 Tape	CH = 31	UN = 40,41 EQ = 5
		885 Disk	CH = 1	
		580 Printer	CH = 12	
41	≥1000K	639 Tape	CH = 6	UN = 0,1
		834 Disk	CH = 0	
		ASY Printer ¹		

1. ASY printer is an asynchronous printer connected to a 255x NPU, a CDCNET MTI, or a CDCNET MDI/TDI.

Table 2-4. Preconfigured Deadstart Decks (NOS Non-810/830 Decks)

Deck Number	Memory Size	ESM Size	Disk Type	UN
07	262K		844	0,1
10	262K		885-12	40,41
11	512K		844	0,1
12	512K		885-12	40,41
13	512K		895	40,41
14	1000K		844	0,1
15	1000K		885-12	40,41
45	1000K		887	0,1
16	1000K		895	40,41
47	1000K		9853	0,1
17	2000K		885-12	40,41
46	2000K		887	0,1
20	2000K		895	40,41
21	262K	1000K	844	0,1
22	262K	1000K	885-12	40,41
23	262K	1000K	885-42	40,41
24	262K	2000K	844	0,1
25	262K	2000K	885-12	40,41
26	262K	2000K	885-42	40,41
27	>=512K	1000K	844	0,1
30	>=512K	1000K	885-12	40,41
31	>=512K	1000K	885-42	40,41
32	>=512K	2000K	844	0,1
33	>=512K	2000K	885-12	40,41
34	>=512K	2000K	885-42	40,41

All non-810/830 decks contain two 67x tape drives (units 0 and 1) on channel 13. The printer is a 580 on channel 12, equipment 5.

Table 2-5. Preconfigured Deadstart Decks (Dual State Decks)

Deck Number	Mainframe Type	Memory Total	NOS Memory (MBs)	UEM Memory (MBs)	VE Memory (MBs)	Disk Type
42	810/830	12	3.5	2	6.5	836
43	Non-810/830	16	6	2	8	885-12
44	Non-810/830	32	4	4	24	895

Step 2: Deadstart the System

The CYBER Initialization Package (CIP) tape contains hardware and software interface modules that allow NOS to run on a CYBER mainframe. CIP is delivered with your NOS order. 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 customer engineer (CE) to ensure that CIP has been installed; CIP installation should be a cooperative effort between you and your CE.

Using the preconfigured deadstart decks you selected in Step 1 and the released deadstart tape, deadstart the system following the instructions given below. For more detailed information on the deadstart process, refer to the NOS Version 2 Operations Handbook; for details on the deadstart panel settings, refer to the CIP User's Handbook.

1. Mount the released 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 CYBER 800 series computers or non-800 series computers where CIP has been installed, set up the panel for a disk deadstart from the disk unit that contains the 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.
 - On the CC545 console, press the deadstart button.
 - On the CC598B console, press CTRL F2.
 - On the CC634B console, press the RESET key to reinitialize the console. Next, press CTRL G and then press CTRL R.
4. Advance to the Initial Options display.
 - For models with a deadstart panel display, press CR.
 - For models 810, 830, 840A, 850A, 860A, 960, 970, 990A, 994, and 995A, the Initial Deadstart display appears after you initiate the deadstart sequence. If you select S on this display, the Initial Options display appears.
 - For all models except 810, 830, 840A, 850A, 860A, 960, 970, 990A, 994, and 995A, the Initial Options display appears after you initiate the deadstart sequence.
5. Check the deadstart panel settings.
 - a. Enter O to select operator intervention.
 - b. Toggle the deadstart state to NOS deadstart.
 - c. Enter P to check the deadstart panel settings. Set the DISPLAY CMRDECK option to YES.
 - d. Press the backspace key to return to the O display.

Step 2: Deadstart the System

6. Advance to the CMRDECK display.
 - For all CYBER 800 series computers and models 960, 970, 990, 994, and 995, enter S to select the deadstart device 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.
7. Enter CMRDECK entries.
 - a. An instructional display (CMRINST) gives the format for CMRDECK entries. To advance to the screen for entering data:
 - On the CC545 console, press the right blank key.
 - On the CC598B console, press the tab key.
 - On the CC634B console, press the right tab key.
 - b. Enter any CMRDECK changes. When you have finished, enter:
NEXT.
8. Enter EQPDECK entries.
 - a. An instructional display (EQPINST) gives the format for EQPDECK entries. To advance to the screen for entering data:
 - On the CC545 console, press the right blank key.
 - On the CC598B console, press the tab key.
 - On the CC634B console, press the right tab key.
 - b. Enter EQPDECK entries to describe your hardware configuration. When you have finished, 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 3: Install Permanent Files

The permanent files for your system are released on the permanent file tapes. You install these files by using a SYSGEN command. SYSGEN automatically creates initial validation files for your system and loads files from the permanent file tapes to the various user names required for system operation. In addition to files required for system operation, initial configuration files for NAM/CCP and NAM/CDCNET networks, online manuals and help files are also loaded to disk.

To install all permanent files, follow these steps:

1. From the system console, 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, you will see the following message in the system dayfile:

```
***** E N D   S Y S G E N *****
```

All the necessary permanent files are set up on your default family device.

Step 4: Bring Up a Network Terminal

When you installed the permanent files for the system, the files necessary for configuring a single line were automatically loaded on the system. Control Data has two networks:

- NAM/CDCNET
- NAM/CCP

Read the hardware requirements pertaining to the network you have installed. Then, follow the steps for activating the network installed on your system.

NAM/CDCNET (DI Requirements and Information)

Here are the minimum hardware requirements for configuring one line:

- A CDCNET Device Interface (DI) configured as a Mainframe Terminal Interface (MTI). An MTI contains lines and terminals and is connected to the mainframe via a CYBER channel.
- Two CDCNET DIs: one configured as a Mainframe Device Interface (MDI) and one configured as a Terminal Device Interface (TDI). An MDI contains an Ethernet cable and is connected to the mainframe by a CYBER channel. It has no lines or terminals. A TDI contains lines and terminals and is connected to an Ethernet cable, not a CYBER channel.

The MTI or TDI must contain an asynchronous line connected to LIM 0, PORT 0. That line and the terminal connected to it have been pre-defined in CDCNET configuration files and are used to complete the installation and configuration of the system. If no line is connected to LIM 0 PORT 0, either physically move a line there or once the DI is loaded, use the Network Operator Utility (NETOU) via its K display to define the line. NETOU is described in the CDCNET Network Operations Manual.

To use the terminal, you must inform the CDCNET software of the system identifiers (serial numbers) of your DIs. Do this by entering the following commands:

1. Define the system IDs of each DI in the network using procedure ADDDI (ADD_DEVICE_INTERFACE). Enter the following commands from the system console:

```
X.DIS.  
USER,NETADMIN,NETADMIN.  
BEGIN,ADDDI,DCNPLIB,type,si.
```

Parameter	Description
type	Type of DI (MDI or MTI).
si	Last 6 digits of the DI's system identifier. The DI system identifier is a 12-digit number that can be found printed on a tag attached to the inside of the DI cabinet. Note that the tag contains a 4-digit checksum appended to the 12-digit system identifier. Do not include the checksum in this parameter.

2. If you configured an MDI in step 1, perform this step; if you configured an MTI, skip to step 3. Execute the ADDDI procedure again to define your TDI. Specify the DI's system identifier (si):

```
BEGIN, ADDDI, DCNPLIB, TDI, si.
```

3. Terminate the DIS package:

```
DROP.
```

To use the terminal, initiate the network by following the steps in *Activating the Network*, described later in this section.

NAM/CCP (255x Requirements and Information)

The minimum hardware requirements for configuring one line are a 65K NPU connected via a channel to your mainframe. The NPU should contain at least one 2561-x CLA for use with an asynchronous line and terminal. The terminal line should be connected to the CLA and the CLA dialed to PORT 0A(16) to run at 600-9600 baud or to PORT 10(16) to run at 110-2400 baud.

To use the terminal, initiate the network by following the steps in *Activating the Network*, described next.

Activating the Network

To activate the network, perform the following steps:

1. Make sure the EST ordinal number for either your 255x NPU (EST number 60) or your MDI or MTI (EST number 61) is turned on. This can be determined by checking the E, display (an NPU is type NP; a DI is type ND). If the one you are to use is off, turn it on by entering this command at the system console:

```
ON, EQ=xx.
```

Replace xx with the EST number for the 255x NPU or CDCNET DI.

2. Reset all DIs or NPUs to ensure the software is loaded with the proper configuration.
3. Initiate NAM and IAF by entering:

```
NAM.  
IAF.
```

It will take several minutes to ready the subsystems and load the 255x or DI hardware. NAM will submit numerous other applications to aid in configuring and loading the network.

4. Log in to the system. For CDCNET networks, you must first create a connection to the default NOS system name. To do so, use the following command:

```
CREC NOS_SYSTEM
```

Step 5: Configure the System

During this step, you will create a new deadstart file (or tape) containing a permanent copy of your deadstart decks; you will also create and/or modify your configuration files.

Deadstart Decks

Before creating deadstart decks, refer to table 2-1 for the list of products that require deadstart deck entries. Chapter 7 gives more detailed information about product deadstart deck entries.

Perform the following steps from an interactive terminal logged in to the installation user name. The default is INSTALL. For a list of default system passwords, refer to SYSGEN Validations in chapter 8. You will need an unlabeled scratch tape for the new deadstart tape.

1. Get a copy of the deadstart decks (CMRDECK, EQPDECK, and so forth) you used in Step 2. Use the following commands:

```
COMMON, SYSTEM.  
GTR, SYSTEM, DSTDECK.deckname1, . . . , decknamen
```

Replace `deckname1` through `decknamen` 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 (such as FSE or XEDIT) to make any necessary changes to the deadstart decks on file `DSTDECK`.
3. When you have made all the necessary changes to the deadstart decks, use the following command to merge the records on file `DSTDECK` with the contents of your current system:

```
SYSGEN, DST, SYSTEM, DSTDECK, NEW, 0, density.
```

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

4. When a tape request is issued, mount an unlabeled scratch tape. Then enter this command at the system console to assign the tape:

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

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

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

Configuration Files

After you have created your new deadstart tape, you should create the configuration files needed to operate your system. Table 2-2 lists the products that require configuration file entries. Refer to chapter 7 for information on creating and moving the configuration files.

Step 6: Deadstart the New System

1. When you have moved all the configuration files to the proper user names, terminate all NAM and IAF subsystem activity. To do so, enter the following commands from the system console:

```
K, NAM.  
K. AP=NVF.  
K. ID, HOST.  
IDLE, IAF.
```

2. Wait for all system activity to complete. Then perform a permanent file backup (PFDUMP) if desired. Consult the NOS Version 2 Analysis Handbook for information on the PFDUMP utility.

3. Shut down the system by entering the following commands from the system console:

```
UNLOCK.  
CHECK POINT SYSTEM.
```

4. Deadstart the new system using the deadstart tape you built in Step 5. Follow the directions for deadstarting the system given in Step 2. Because the default family device contains the permanent files needed to run the system, do not initialize that device.

Step 7: Wrap Up

You have now completed the first-time installation process. If you need to customize the system, go to chapter 6 for more installation information. If you are installing a dual state system, go to the NOS/VE Installation Handbook. The following three topics contain suggestions that you may wish to consider.

User Names and Passwords

For security reasons, it is suggested that the passwords for the system user names be changed upon completion of the installation. Also, SYSGEN automatically created user names for all products; you can delete the user names of products your site did not install. Refer to SYSGEN Validations in chapter 8 for information on modifying passwords.

You should also create user names and passwords for the users of the system. Refer to the NOS Version 2 Administration Handbook for information on creating user names and passwords.

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. You can access the bound version of the online manuals with 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. 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 was automatically installed as file CML3 to user name CDCCE by the X.SYSGEN(FULL) command executed in Step 3. 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 installation is complete.

Introduction	3-1
Step 1: Create New User Names	3-2
Step 2: Set Up Installation Tools	3-3
Step 3: Update Decks and Configuration Files	3-4
Updating the Deadstart Decks	3-5
Creating or Updating Configuration Files	3-5
Step 4: Write a New Deadstart Tape	3-6
Step 5: Override Automatic File Installation	3-7
Step 6: Install Permanent Files	3-12
Step 7: Deadstart the New System	3-13

Introduction

The steps in this chapter describe how to install a new release of NOS on a system that is currently running a prior release of NOS. If you are upgrading a dual state system, follow these same instructions.

The installation steps are designed to allow you to perform as much of the upgrade as possible on your existing NOS system before you deadstart the new release tape. Most of the steps can be performed from an interactive terminal running on your current production system.

An upgrade installation consists of the steps listed below. These steps are described in the remaining sections of this chapter. If you are installing any new products with this upgrade, you will find it helpful to review the steps in chapter 2, First-Time Installation.

1. Create New User Names
2. Set Up Installation Tools
3. Update Decks and Configuration Files
4. Write a New Deadstart Tape
5. Override Automatic File Installation
6. Install Permanent Files
7. Deadstart the New System

Consult the Software Release Bulletin (SRB) and any Installation Bulletins (IBs) for important notes or cautions before upgrading to the new release of NOS.

Step 1: Create New User Names

For each product that you are installing for the first time, create the user names necessary for running the product. By creating the user names now, you can install files to the user names before deadstarting the new system. To determine the new user names, refer to these sources:

- The SRB lists any user names that are new for this release.
- Table 8-1 (in chapter 8) lists all of the user names on which the system will store files and the products that use those user names.

If you wish to change the default Control Data installation user names to ones that are site-defined, the site user names must be created now. The user names involved are the installation user name (default INSTALL), the network operations user name (default NETOPS), and the network administration user name (default NETADMN). Please note that the user names you choose should have similar validations to the default values supplied by Control Data.

Any new user names must be created/updated from the system console using the MODVAL utility. Refer to the NOS Version 2 Administration Handbook for details about MODVAL.

In addition to creating the new user names, you must update your ZZSYSGU file that SYSGEN uses for executing USER commands. SYSGEN expects file ZZSYSGU to be present on user name SYSTEMX. If the NOS system you are upgrading is a pre-level 664/650 system or file ZZSYSGU is not present on user name SYSTEMX, perform step 1; otherwise skip to step 2.

1. Copy the released deadstart tape to a local file named SYSTEM and execute the SYSGEN command that saves file ZZSYSGU on user name SYSTEMX.

```
X.DIS.  
USER, SYSTEMX, password.  
REQUEST, TAPE, VSN=vsn, D=density, LB=KU, F=I, PO=R.  
COPYEI, TAPE, SYSTEM, V.  
UNLOAD, TAPE.  
GTR, SYSTEM, PFG.ULIB/PFGLIB  
GTR, SYSTEM, SYSGEN.PROC/SYSGEN  
BEGIN, SYSGEN, SYSTEM, LOADUSE.
```

Replace password with the password for the SYSTEMX user name; replace vsn with the VSN of the released deadstart tape; and replace density with the density of that tape.

2. Modify ZZSYSGU to contain the correct user names and passwords for your site. Refer to SYSGEN Validations in chapter 8 for more information.

Step 2: Set Up Installation Tools

This step sets up the RECLAIM database and a global library of tools to help recompile configuration files. By setting up the RECLAIM database, you can perform much of the upgrade work before deadstarting the new system.

You will need the new released deadstart tape. If you are customizing, you will also need the new released permanent file tapes. Perform the following steps:

1. Log in to your installation user name using an interactive terminal.
2. Copy the new released deadstart tape to a local file named SYSTEM:

```
REQUEST, TAPE, VSN=vsn, D=density, LB=KU, F=I, PO=R.
```

Replace `vsn` with the VSN of the released deadstart tape (contained in the media number field of the external tape label). Replace `density` with the tape density of the tape (HY, HD, PE, or GE).

```
COPYEI, TAPE, SYSTEM, V.
UNLOAD, TAPE.
```

NOTE

Execute the instructions in chapter 6 if you wish to do any of the following tasks:

- Customize NOS or its products
- Install a newer version of NOS with an older version of NOS/VE (for example, NOS L688 with NOS/VE L678)
- Change the default Control Data installation user names (INSTALL, NETOPS, and NETADMN) to ones defined by the site

Pay close attention to the introductory notes in chapter 6. When you have completed the steps in chapter 6, continue the upgrade with Step 3: Update Decks and Configuration Files, in this chapter.

3. Set up the RECLAIM database and global library file by entering the following commands:

```
GTR, SYSTEM, PFG.ULIB/PFGLIB
BEGIN, SYSGEN, SYSTEM, INIT, UPGRADE.
```

When this procedure is completed, you have created the following files:

- RECLDB, a new RECLAIM database, containing information about all the files on the new released permanent file tapes. Any previous copy of file RECLDB is replaced.
- PRODUCT and GLOBLIB. PRODUCT contains the SYSGEN procedures used to install NOS and its products. GLOBLIB contains the programs and procedures used to recompile configuration files and install new products for the first time.

Step 3: Update Decks and Configuration Files

The programs and procedures in GLOBLIB allow you to execute newly released versions of the following utilities:

Utility	Description
ANACD	Dump Analyzer Utility. Used for analyzing CDCNET access dumps. Before using ANACD, you must install the new version of CDCNET to disk.
MANCC	CDCNET Configuration Utility. Used for managing CDCNET configuration procedures. Before using MANCC, you must install the new version of CDCNET to disk.
NDLP	Network Definition Language Processor. Used for compiling Network Definition Language (NDL) files to produce a new LCFFILE and NCFFILE.
NETFM	Network File Manager. Used for accessing the CDCNET Network Directory file (NETDIR).
NETPLM	Network Procedure Library Manager. Used for maintaining libraries of CDCNET configuration procedure libraries.
NPA	Network Performance Analyzer. Used for analyzing CDCNET performance. Before using NPA, you must install the new version of CDCNET to disk.
RCFGEN	RHF Configuration File Generator. Used for compiling a new RCFMid file for RHF.
SYSGEN	System Generator. Used for loading permanent files from the released permanent file tapes and installing them on the system.

To use the programs loaded into GLOBLIB, enter the following commands:

```
ATTACH, GLOBLIB/UN=insun.  
LIBRARY, GLOBLIB/A.
```

Step 3: Update Decks and Configuration Files

During this step you will:

- Update the deadstart decks
- Create or update configuration files

Do this from an interactive terminal logged in to the installation user name for most files. The NDL file and files belonging to CDCNET must be updated on the network administration user name.

Updating the Deadstart Decks

Modify the existing CMRDECKs and EQPDECKs. Include additional entries for any new products that require entries and change the CMRDECK VERSION entry to the new NOS release. Place the updated decks on file DSTDECK. Refer to table 3-1 and chapter 7 for information about each product.

To update IPRDECKs and LIBDECKs, either add any local entries to the released decks or modify your local decks to include the new entries in the released decks. To get a copy of the released decks, first refer to Initial Deadstart Preparation in chapter 2 to determine which released IPRDECK and LIBDECK to use. Select the decks that match your type of mainframe and extended memory usage. Then, get the desired decks from the new released deadstart tape. Add the updated decks to file DSTDECK. Save file DSTDECK as a permanent file for use during Step 4.

Creating or Updating Configuration Files

To update or create configuration files, refer to table 2-2 for a list of the products that require configuration file entries. Note the following:

- If you are installing a product for the first time and the product requires configuration file entries, refer to chapter 7 for more first-time configuration information for each product.
- Regardless of whether you need to update files LCFFILE, NCFFILE, and RCFMid, you must recompile these files when you upgrade the network. Refer to the NAM5 section of chapter 7 for information about files LCFFILE and NCFFILE and to the RHF section of chapter 7 for information about file RCFMid.
- If you are installing CDCNET for the first time as part of your upgrade installation, execute the instructions in the Installing CDCNET for the First Time section of chapter 7.
- If you are upgrading CDCNET as part of your upgrade installation, execute the instructions in the Upgrading CDCNET section of chapter 7.
- Configuration files that are eventually stored on special system user names (such as SYSTEMX, the network operations user name, and SUBFAM0) should be temporarily stored on the installation user name. SYSGEN(MOVE) is used to place these files on the correct user names during Step 6: Install Permanent Files. Do not move configuration files until Step 6.
- Configuration files that are created or updated on the network administration user name (for example, NDL files, CCPLOAD files, and CDCNET configuration files) should be stored with temporary file names to avoid conflicts with the running system.

Step 4: Write a New Deadstart Tape

To write a new deadstart tape, use the new released deadstart tape as a base and add your own local deadstart decks and the binaries for any customized products. You will need an unlabeled scratch tape for the new deadstart tape.

Perform the following steps from an interactive terminal logged in to the installation user name.

1. If the new released deadstart tape is on local file SYSTEM, skip this step. Otherwise, enter the following commands:

```
REQUEST, TAPE, VSN=vsn, D=density, LB=KU, F=I, PO=R.  
COPYEI, TAPE, SYSTEM, V.  
UNLOAD, TAPE.
```

Replace `vsn` with the VSN of the released deadstart tape (contained in the media number field of the external tape label). Replace `density` with the tape density of the tape (HY, HD, PE, or GE).

2. Make file DSTDECK (created in Step 3) a local file.
3. If you did not customize binaries, place any LIBEDIT directives that are necessary to add your local deadstart decks to the deadstart tape on local file USERD. If no LIBEDIT directives are necessary, use a 0 in place of the USERD in step 4.

If you customized binaries, place any necessary LIBEDIT directives on local file USERD. File USERD should contain a *FILE,DSTDECK directive to reference the deadstart decks in file DSTDECK.

4. Create the new deadstart tape.

- If you did not customize binaries, create the new deadstart tape using these commands:

```
RETURN, NEW.  
SYSGEN, DST, SYSTEM, DSTDECK, NEW, USERD, density.
```

Replace `density` with the density of the new deadstart tape (HY, HD, PE, or GE). A tape request will be issued for an unlabeled tape with VSN=NDT. The tape may be assigned from the system console using the following command:

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

Replace `est` with the EST ordinal number of the tape unit where the new deadstart tape is mounted. LIBEDIT output is written to a local file named SYSLIST.

5. If you did customize deadstart binaries, create the new deadstart tape using the following commands. Local file USERD (if created in step 3) provides any necessary LIBEDIT directives.

```
RETURN, NEW.
BEGIN, GENDST, INSTALL, D=density, LIST=list.
```

Replace density with the density of the new deadstart tape (HY, HD, PE, or GE). Replace list with the name of the file to receive the LIBEDIT output. A tape request will be issued for an unlabeled scratch tape with VSN=NDT. The tape may be assigned from the system console using the following command:

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

Replace est with the EST ordinal number of the tape unit where the new deadstart tape is mounted.

Step 5: Override Automatic File Installation

At this point in the installation, you have completed the following tasks:

- Customized any other released permanent files as described in Customizing Permanent Files in chapter 6 (refer to Step 2).
- Created all new configuration files and updated all existing configuration files as described in Step 3 of this chapter.

Before you use SYSGEN to load permanent files (Step 6), you must first examine table 3-1 to determine which files you do not want to load from the release tapes. The following files should not be loaded:

- Files PFGDCNS and PFGCHA2, since CDCNET is installed using instructions in chapter 7. Do not prevent PFGCHA1 or PFGTCPH from loading; they are needed to update NAMSTRT.
- Any configuration files already on your system (such as LCFFILE, LIDCMid, LIDVEid, NCFFILE, and NDLDATA).
- Any files you customized when building products from source in chapter 6 (such as NLFFILE).
- Any files on your system that you do not want replaced by new released versions (for example, subsystem startup procedures).

Table 3-1 is organized by product name and lists the user names and file names associated with each product. The right column lists the PFGxxx files associated with each product. Make a list of all the files you do not want loaded to the new system.

The following steps prevent files from loading by making temporary modifications to the RECLAIM database so that the selected PFGxxxx files appear not to be on the permanent file tape. That is, the file names are deleted from the database.

After listing the files you do not want loaded to the new system, perform the following steps from a terminal logged in to the installation user name:

1. Execute the RECLAIM utility by entering this command:

```
RECLAIM.
```

RECLAIM responds with the following prompt:

```
ENTER DIRECTIVE.  
?
```

2. Enter the DELETE directive to temporarily disable processing of the specified files:

```
DELETE,PF=*,UN=NS2psrin
```

Replace psrin with the PSR level of the NOS release you are installing. RECLAIM prompts you for a list of file names:

```
ENTER FILE NAMES.  
?
```

3. Enter the desired file names:

```
filename1,...,filenamem
```

Separate each file name with a comma. If it is not possible to fit all the file names on one line, follow the last file name with a comma and continue with more file names at the next prompt. After you type the last file name, enter a CR. Do not end the line with a period or a comma.

RECLAIM responds with a report of the file names that have been deleted, then displays the following prompt:

```
ENTER DIRECTIVE.  
?
```

4. Terminate the RECLAIM session by entering the following:

```
END
```

RECLAIM responds with the following message:

```
RECLAIM COMPLETE.
```

Here are some additional notes on using RECLAIM:

- You can list the names of all deleted files by executing the following command:

```
RECLAIM,Z./LIST,UN=NS2psrin,DE
```

- You can restore the files back to active status by executing the following command:

```
RECLAIM,Z./RESET,UN=NS2psrin,PF=*/filename1,...,filenamem
```

- If you have only a few files to load, you can delete all the files from the database, then restore only the desired files to active status. For example:

```
RECLAIM,Z./DELETE,UN=NS2psrin  
RECLAIM,Z./RESET,UN=NS2psrin,PF=*/filename1,...,filenamem
```

Table 3-1. Automatic File Installation

Product Name	User Name	File Name	PFGxxxx Name
APL 2	APL0	1	PFGAPL2
	APL1	1	
ATF	netopun ³	NAMSTR2 ²	PFGATF1
	APL1	1	
Communications Control Program (CCP) 3	netadun ³	NLFFILE	PFGCCPL
Concurrent Maintenance Library (CML)	CDCCE	CML3	PFGCML1
Control Data Distributed Communications Network (CDCNET)	netadun ³	1	PFGDCNS
	netopun ³	NETDIR	
	insun ³	1	PFGCHA1,
	netadun ³	1	PFGCHA2
CYBER Database Control System 2	netopun ³	NAMSTR2 ²	
	netopun ³	NAMSTR2 ²	
CYBER Database Control System 2	SYSTEMX	CDC	PFGDCS
Data Catalogue 2 Version 2.0	LIBRARY	1	PFGDCL2
Dual State	SYSTEMX	LIDCMid, LIDVEid, VEMEM	PFGDUAL
	insun ³		
Full Screen Editor	SYSTEMX	SMF	PFGFSE1
	insun ³	SMFSTAT	
	LIBRARY	FSEHELP,	
		FSEPROC, FSTEACH	
Interactive Facility (IAF) 1	SYSTEMX	IAF, IAFTM, IAFTR	PFGIAF1

1. Numerous files are installed with this product. Typically, these files are handled as a group and are therefore not listed here. Refer to chapter 7 and appendix B for more information.

2. The NAMSTR2 file on the network operations user name is updated by multiple products. The NAMSTR2 file is initially created by NAM and is modified by other products.

3. insun, netopun, and netadun refer to the Control Data default values for the installation user name, network operations user name, and the network administration user name. The default values are INSTALL, NETOPS, and NETADMN, respectively. If you have changed these values, substitute your site names for the Control Data default values.

(Continued on next page)

Table 3-1. Automatic File Installation

Product Name	User Name	File Name	PFGxxxx Name
<i>(Continued from previous page)</i>			
Interactive Transfer Facility (ITF)	netopun ³	NAMSTRT ²	PFGITF1
Maintenance Package	SYSTEMX	SECART, MSGID	PFGTOOL
MAP Macro Control Library (MMCL) V3	SYSTEMX	MAPLIBC, MAPLIBE	PFGMMCL
Mass Storage Extended (MSE)	SYSTEMX SUBFAM0	MSE, MSESLAV MSECONF	PFGMSE1
Message Control System (MCS) 1	SYSTEMX	MCS	PFGMCS1
MSSI V3, MAP III	SYSTEMX	MAPCH, MAPECS, MAPCMI	PFGMSSI
	insun ³	MSSIP, MJOBSPL	
	CDCCE	1	PFGAP1I,
	CDCCE2	1	PFGAP1L
Network Access Method (NAM) 1	SYSTEMX	NAM	PFGNAM5
	netopun ³	NAMNOGO	
	netadun ³	NAMSTRT ² LCFFILE, NCFFILE, NDLDATA	PFGNDL1
NOS Online Manuals	MANUALS	1	PFGMAN1
NOS Scope Station Facility	SYSTEMX	SSF	PFGNSS1

1. Numerous files are installed with this product. Typically, these files are handled as a group and are therefore not listed here. Refer to chapter 7 and appendix B for more information.

2. The NAMSTRT file on the network operations user name is updated by multiple products. The NAMSTRT file is created initially by NAM and is modified by other products.

3. insun, netopun, and netadun refer to the Control Data default values for the installation user name, network operations user name, and the network administration user name. The default values are INSTALL, NETOPS, and NETADMN, respectively. If you have changed these values, substitute your site names for the Control Data default values.

(Continued on next page)

Table 3-1. Automatic File Installation

Product Name	User Name	File Name	PFGxxxx Name
<i>(Continued from previous page)</i>			
NOS 2 Package	MANUALS	¹	PFGONLM
	LIBRARY	PFTFTR, RMUGET	PFGPFTF
	LIBRARY	TDUFILE, TERMLIB	PFGTLIB
Printer Support Utility (PSU)	LIBRARY	HELPLIB	PFGNOSB
	SYSTEMX	RDF, STAGE	PFGRDF1 PFGNOS2
	netopun ³	EVFULFN, NAMSTR ²	PFGPSU1
PTF/QTF File Transfer Facility	netopun ³	NAMSTR ²	PFGRHP1
Remote Batch Facility (RBF) 1	netopun ³	NAMSTR ²	PFGRBF5
TCP/IP/FTP/TELNET	netopun ³	NAMSTR ²	PFGTCPH
Transaction Facility (TAF) 1	SYSTEMX	TAF	PFGTAF1
	KB100DC	TASKLIB	
XEDIT 3	LIBRARY	XEDITH	PFGXEDT

1. Numerous files are installed with this product. Typically, these files are handled as a group and are thus not listed here. Refer to chapter 7 and appendix B for more information.

2. The NAMSTR file on the network operations user name is updated by multiple products. The NAMSTR file is created initially by NAM and is modified by other products.

3. insun, netopun, and netadun refer to the Control Data default values for the installation user name, network operations user name, and the network administration user name. The default values are INSTALL, NETOPS, and NETADMN, respectively. If you have changed these values, substitute your site names for the Control Data default values.

Step 6: Install Permanent Files

Before you install the permanent files, do the following:

- Ensure that all users have logged off the system.
- Ensure that all normal production activity is terminated.
- Ensure that all subsystem activity is terminated (except for the MAG and BIO subsystems).

Back up the permanent files using the PFDUMP utility. You can use the backup tapes if you need to restore any files. Refer to the NOS Version 2 Analysis Handbook for information about the PFDUMP utility.

To install new permanent files, perform the following steps from the system console.

1. Access the new SYSGEN procedures:

```
X.DIS.  
USER,SYSTEMX,password.  
GET,SYSGEN/UN=insun.  
ATTACH,GLOBLIB/UN=insun.  
LIBRARY,GLOBLIB/A.
```

Replace password with the password for the SYSTEMX user name.

2. Have your permanent file tapes available for mounting and note the VSN of the first permanent file tape. The VSN is in the media number field of the external tape label. Invoke SYSGEN by entering the following command from DIS:

```
SYSGEN,FILES,vsn.
```

Replace vsn with the volume serial number of the first permanent file tape.

3. Press the period key to cause the SYSGEN procedure to begin execution.

SYSGEN loads the files from the permanent file tapes and installs them to their proper user names on the system. After SYSGEN installs the files from the permanent file tapes, you can install the site-modified files created in Step 3. Do this by executing steps 4 and 5 below as many times as necessary to move all site-modified files.

4. Reset the user name of your DIS package back to user name SYSTEMX:

```
USER,SYSTEMX,password.
```

Replace password with the password for the SYSTEMX user name.

5. Move a file from the installation user name to its proper location by using the SYSGEN(MOVE) function. Refer to chapter 8 for more information about calling SYSGEN and SYSGEN(MOVE).

Refer to table 3-1 to determine the proper user name for each file. If you did not create or modify any configuration files in Step 3, SYSGEN(MOVE) does not need to be executed.

6. If you need to set any file permissions, do so after the SYSGEN (MOVE) command is completed.

If in Step 3 some files were created with temporary names (for example, the NDL files), change them to their correct file names.

If you recompiled files LCFFILE and NCFFILE, follow the instructions for upgrading the files in the NAM5 section of chapter 7.

If you recompiled file RCFMid, follow the instructions for upgrading the file in the RHF section of chapter 7.

If you changed the default Control Data installation user names (INSTALL, NETOPS, and NETADMN) to ones that are site-defined, note the information given in Site-Defined User Names in the NAM5 section of chapter 7

If you installed an upgrade to CDCNET, follow the instructions in Activating a New Level of CDCNET in the CDCNET section of chapter 7.

Step 7: Deadstart the New System

Before deadstarting the new system, be sure that the corresponding level of CIP for this NOS release has been installed. Refer to the CIP User's Handbook and the CIP SRB for more information.

Deadstart the system using the new deadstart tape. When the system comes up, it will be using all of the newly released software and permanent files. The new system is now ready for production. If you installed an upgrade of CDCNET, you may want to perform the cleanup activity described in the CDCNET section of chapter 7.

Your upgrade installation is now complete.

Component Order Installation

- Introduction 4-1
- Step 1: Create New User Names 4-2
- Step 2: Set Up Installation Tools 4-2
 - Products Requiring Configuration Tools 4-3
 - Permanent File Based Products 4-4
 - Standard Products 4-5
- Step 3: Update Configuration Files and Decks 4-6
- Step 4: Write a New Deadstart Tape 4-7
- Step 5: Override Automatic File Installation 4-8
- Step 6: Install Permanent Files 4-10
- Step 7: Deadstart the 4-11

Introduction

This chapter describes how to install additional products on a system currently running NOS. The PSR level of the components you want to install must be the same as that of the NOS software running on your system.

NOTE

It is assumed that sites will use the same installation user names that were used to install the currently running NOS operating system.

A component order is an order that does not include NOS. For a component order, you receive a set of permanent file tapes; you do not receive a new deadstart tape. All the necessary binaries for your order are on the permanent file tapes.

A component order installation consists of the steps listed below. These steps are described in the remaining sections of this chapter:

1. Create New User Names
2. Set Up Installation Tools
3. Update Configuration Files and Decks
4. Write a New Deadstart Tape
5. Override Automatic File Installation
6. Install Permanent Files
7. Deadstart the New System

NOTE

If you have CDCNET already installed and now need to install a CDCNET separate TIP as part of a component order, follow the instructions given in Installing a CDCNET Separate TIP in the CDCNET section of chapter 7. If your component order contains products in addition to a CDCNET separate TIP, you must still execute the instructions in chapter 4 after you have installed the TIP.

TCP/IP/FTP/TELNET consists of a CDCNET separate TIP and host applications. For this reason, you should first perform the instructions given in Installing a CDCNET Separate TIP in the CDCNET section of chapter 7, and then perform the instructions in chapter 4 to install the host applications.

Step 1: Create New User Names

For each product you install, create the user names necessary to run the product. By creating the user names at this time, you can install files to the user name before deadstarting the new system. To determine the new user names, refer to table 8-1 in chapter 8, which lists all the user names on which the system stores files, along with the products that use these user names.

If you wish to change the default Control Data installation user names to ones that are site-defined, the site user names must be created now. The user names involved are the installation user name (default INSTALL), the network operations user name (default NETOPS), and the network administration user name (default NETADMN). Please note that the user names you choose should have similar validations to the default values supplied by Control Data.

Any new user names must be created/updated from the system console using the MODVAL utility. Refer to the NOS Version 2 Administration Handbook for details about MODVAL.

In addition to creating the new user names, you must update your ZZSYSGU file (which SYSGEN uses for executing USER commands) to contain the correct user names and passwords for your site. Refer to SYSGEN Validations in chapter 8 for more information.

Step 2: Set Up Installation Tools

To set up the installation tools for the products you are installing, follow only the instructions that apply to the products for your site:

- If your component order includes one of the following products, follow the instructions in this step titled Products Requiring Configuration Tools:
 - Control Data Distributed Communications Network (CDCNET)
 - Remote Host Facility (RHF)
- If your component order consists of only the following products, follow the instructions in this step titled Permanent File Based Products:
 - Communications Control Program (CCP) 3
 - Conversion Aids Subsystems 3
 - CROSS
 - Data Catalogue 2 Version 2.0
 - Link Interface Program under CCP 3
 - NOS Online Manuals
 - 3270 TIP for CCP
- If none of the products listed above are in your component order, follow the instructions in this step titled Standard Products.

Products Requiring Configuration Tools

If your component order includes either RHF or CDCNET, you must load configuration tools to the global library file GLOBLIB on the installation user name. To do so, perform the following steps. Before beginning, ensure that the released permanent file tapes are available for mounting.

1. Log in to your installation user name on an interactive terminal.
2. Access the current system to use as a base for the SYSGEN(UPGRADE) procedure:

```
COMMON, SYSTEM.
```

3. Execute the following command to merge the binaries on the component order tape with file SYSTEM:

```
SYSGEN, UPGRADE, SYSTEM, NEWSYS, vsn, density.
```

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

This command updates the RECLAIM database, RECLDB, with information about the files on the permanent file tapes. The command also produces a new deadstart file named NEWSYS which contains all of the programs and binaries from your current system and those from the permanent file tapes.

4. Place the binaries of the CDCNET and/or RHF configuration tools into your global library file GLOBLIB:

```
RENAME, SYSTEM=NEWSYS.  
SYSGEN, INIT, SEED.
```

To use the programs loaded into GLOBLIB, use the following commands:

```
ATTACH, GLOBLIB.  
LIBRARY, GLOBLIB/A.
```

NOTE

Execute the instructions in chapter 6 if you wish to do either of the following tasks:

- Customize NOS or its products
- Change the default Control Data installation user names (INSTALL, NETOPS, and NETADMN) to ones defined by the site

Pay close attention to the introductory notes in chapter 6. When you have completed the steps in chapter 6, continue the upgrade with Step 3: Update Configuration Files and Decks in this chapter.

If you don't need to execute the instructions in chapter 6, continue with Step 3: Update Configuration Files and Decks in this chapter.

Permanent File Based Products

If your component order consists only of products that do not add binaries to the deadstart tape, perform the following steps. Before beginning, ensure that the released permanent file tapes are available for mounting.

1. Log in to your installation user name using an interactive terminal.
2. Enter this command:

```
SYSGEN,UPGRADE,0,0,vsn,density.
```

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

This procedure updates the RECLAIM database, RECLDB, on your installation user name.

NOTE

Execute the instructions in chapter 6 if you wish to do either of the following tasks:

- Customize NOS or its products
- Change the default Control Data installation user names (INSTALL, NETOPS, and NETADMN) to ones defined by the site

Pay close attention to the introductory notes in chapter 6. When you have completed the steps in chapter 6, continue the upgrade with Step 3: Update Configuration Files and Decks in this chapter.

If you don't need to execute the instructions in chapter 6, go to Step 5: Override Automatic File Installation in this chapter.

Standard Products

If your component order does not contain any of the products listed at the beginning of this step, perform the following steps. Before beginning, be sure the released permanent file tapes are available for mounting.

1. Log in to your installation user name on an interactive terminal.
2. Access the current system to use as a base for the SYSGEN(UPGRADE) procedure:

```
COMMON, SYSTEM.
```

3. Execute the following command to merge the binaries on the component order tape with file SYSTEM:

```
SYSGEN, UPGRADE, SYSTEM, NEWSYS, vsn, density.
```

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

This command updates the RECLAIM database, RECLDB, with information about the files on the permanent file tapes. The command also produces a new deadstart file named NEWSYS, which contains all of the programs and binaries from your current system and those from the permanent file tapes.

4. Make file SYSTEM the merged deadstart file:

```
RENAME, SYSTEM=NEWSYS.
```

NOTE

Execute the instructions in chapter 6 if you wish to do either of the following tasks:

- Customize NOS or its products
- Change the default Control Data installation user names (INSTALL, NETOPS, and NETADMN) to ones defined by the site

Pay close attention to the introductory notes in chapter 6. When you have completed the steps in chapter 6, continue the upgrade with Step 3: Update Configuration Files and Decks in this chapter.

If you don't need to execute the instructions in chapter 6, this file will be used in Step 4 to create a new deadstart tape. Proceed to Step 3: Update Configuration Files and Decks in this chapter.

Step 3: Update Configuration Files and Decks

During this step you should:

- Update the deadstart decks. Table 2-1 lists the products that require deadstart deck entries. For more information about each product, refer to chapter 7.
- If you are installing CDCNET for the first time or upgrading CDCNET, execute the steps in the CDCNET section of chapter 7 that pertain to your type of CDCNET installation.
- Update (or create) configuration files. Table 2-2 lists products that require configuration file entries. For more information about each product, refer to chapter 7.

NOTE

Configuration files that are eventually stored on special system user names (such as the network operations user name, SUBFAM0, and SYSTEMX) should be temporarily stored on the installation user name. SYSGEN(MOVE) will be used to place these files on the correct user names during Step 6: Install Permanent Files. Do not move configuration files until Step 6.

Step 4: Write a New Deadstart Tape

To write a new deadstart tape, use the new file `SYSTEM` you created when following the instructions for either products requiring configuration tools or standard products in Step 2 of this chapter. Add your own local deadstart decks and the binaries for any customized products. You need an unlabeled scratch tape for the new deadstart tape.

Perform the following steps from an interactive terminal logged in to the installation user name.

1. Put the new deadstart decks on the local file `DSTDECK`.
2. If you did not customize any binaries, place any `LIBEDIT` directives necessary to add your local deadstart decks to the deadstart tape on local file `USERD`. If no `LIBEDIT` directives are necessary, use a 0 in place of `USERD` in step 3.

If you customized binaries, place any necessary `LIBEDIT` directives on local file `USERD`. File `USERD` should contain a `*FILE,DSTDECK` directive to reference the deadstart decks in file `DSTDECK`.

3. Create the new deadstart tape.
 - If you did not customize any binaries, create the new deadstart tape using these commands:

```
RETURN,NEW.
SYSGEN,DST,SYSTEM,DSTDECK,NEW,USERD,density.
```

Replace `density` with the density of the new deadstart tape (`HY`, `HD`, `PE`, or `GE`). A tape request will be issued for an unlabeled tape with `VSN=NDT`. The tape may be assigned from the system console using the following command:

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

Replace `est` with the `EST` ordinal number of the tape unit where the new deadstart tape is mounted. `LIBEDIT` output is written to a local file named `SYSLIST`.

- If you did customize any deadstart binaries, create the new deadstart tape using the following commands. Local file `USERD` (if created in step 2) provides any necessary `LIBEDIT` directives:

```
RETURN,NEW.
BEGIN,GENDST,INSTALL,D=density,LIST=list.
```

Replace `density` with the density of the new deadstart tape (`HY`, `HD`, `PE`, or `GE`). Replace `list` with the name of the file to receive the `LIBEDIT` output. A tape request will be issued for an unlabeled scratch tape with `VSN=NDT`. The tape may be assigned from the system console using the following command:

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

Replace `est` with the `EST` ordinal number of the tape unit where the new deadstart tape is mounted.

Step 5: Override Automatic File Installation

At this point in the installation, you have completed the following tasks:

- Customized any other released permanent files as described in Customizing Permanent Files in chapter 6 (refer to Step 2).
- Created all new configuration files and updated all existing configuration files as described in Step 3 of this chapter.

Before you use SYSGEN to load permanent files (Step 6), you must first examine table 3-1 to determine which files you do not want to load from the release tapes. The following files should not be loaded:

- Files PFGDCNS and PFGCHA2, since CDCNET is installed using instructions in chapter 7. Do not prevent PFGCHA1 or PFGTCPH from loading, because they are needed to update NAMSTRT.
- Any configuration files already on your system (such as LCFFILE, LIDCMid, LIDVEid, NCCFILE, and NDLDATA).
- Any files you customized when building products from source in chapter 6 (such as NLFFILE).
- Any files on your system that you do not want replaced by new released versions (for example, subsystem startup procedures).

Table 3-1 is organized by product name and lists the user names and file names associated with each product. The right column lists the PFGxxxx files associated with each product. Make a list of all the files you do not want loaded to the new system.

The following steps prevent files from loading by making temporary modifications to the RECLAIM database so that the selected PFGxxxx files appear not to be on the permanent file tape. That is, the file names are deleted from the database.

After you list the files you do not want loaded to the new system, perform the following steps from a terminal logged in to the installation user name:

1. Execute the RECLAIM utility by entering this command:

```
RECLAIM.
```

RECLAIM responds with the following prompt:

```
ENTER DIRECTIVE.  
?
```

2. Enter the DELETE directive to temporarily disable processing of the specified files:

```
DELETE,PF=*,UN=NS2psrin
```

Replace psrin with the PSR level of the NOS release you are installing. RECLAIM prompts you for a list of file names:

```
ENTER FILE NAMES.  
?
```

3. Enter the desired file names:

```
filename1, ..., filename
```

Separate each file name with a comma. If it is not possible to fit all the file names on one line, follow the last file name with a comma and continue with more file names at the next prompt. When you have typed the last file name, enter a CR. Do not end the line with a period or a comma.

RECLAIM responds with a report of the file names deleted, then displays the following prompt:

```
ENTER DIRECTIVE.  
?
```

4. Terminate the RECLAIM session by entering the following:

```
END
```

RECLAIM responds with the following message:

```
RECLAIM COMPLETE.
```

Here are some additional notes on using RECLAIM:

- You can list the names of all deleted files by executing the following command:

```
RECLAIM, Z./LIST, UN=NS2psrin, DE
```

- You can restore the files back to active status by executing the following command:

```
RECLAIM, Z./RESET, UN=NS2psrin, PF=*/filename1, ..., filename
```

- If you have only a few files to load, you can delete all the files from the database, then restore only the desired files to active status. For example:

```
RECLAIM, Z./DELETE, UN=NS2psrin  
RECLAIM, Z./RESET, UN=NS2psrin, PF=*/filename1, ..., filename
```

Step 6: Install Permanent Files

Before you install the permanent files, do the following:

- Ensure that all users have logged off the system.
- Ensure that all normal production activity is terminated.
- Ensure that all subsystem activity is terminated (except for the MAG and BIO subsystems).

Back up the permanent files using the PFDUMP utility. You can use the backup tapes if you need to restore any files. Refer to the NOS Version 2 Analysis Handbook for information about the PFDUMP utility.

Once you complete the permanent file backup, leave the system running so you can move the new files to their proper user names.

Before beginning, ensure that the permanent file tapes are available for mounting and note the VSN of the first permanent file tape. The VSN is listed in the media number field of the external tape label. To install new permanent files, perform the following steps from the system console:

1. Access the new SYSGEN procedure:

```
X.DIS.  
USER,SYSTEMX,password.  
GET,SYSGEN/UN=insun.  
ATTACH,GLOBLIB/UN=insun.  
LIBRARY,GLOBLIB/A.
```

Replace password with the password for the SYSTEMX user name.

2. Install unmodified permanent files by entering the following command:

```
SYSGEN(FILE,vsn)
```

Replace vsn with the volume serial number of the component order permanent file tape.

3. Press the period key for execution of the SYSGEN procedure. SYSGEN loads the files from the permanent file tape and installs them to their proper user names on the system. After SYSGEN installs the files from the permanent file tape, you can install your site-modified files. Do this by executing Steps 4 and 5 as many times as necessary to install all of the modified files.

4. Reset the user name of your DIS package back to user name SYSTEMX:

```
USER,SYSTEMX,password.
```

5. Move a file from the installation user name to its proper location by using the SYSGEN(MOVE) function. This function must be called using the DSD format. Refer to chapter 8 for more information about calling SYSGEN and SYSGEN(MOVE).

If you did not create or modify any configuration files in Step 3, SYSGEN(MOVE) does not need to be executed.

6. If you need to set any file permissions, do so after the SYSGEN(MOVE) command is completed.

If in Step 3 some files were created with temporary files (for example, the NDL files), change them to their correct file names.

If you recompiled files LCFFILE and NCFFILE, follow the instructions for upgrading the files in the NAM5 section of chapter 7.

If you recompiled file RCFMid, follow the instructions for upgrading the file in the RHF section of chapter 7.

If you changed the default Control Data installation user names (INSTALL, NETOPS, and NETADMN) to ones that are site-defined, note the information given in Site-Defined User Names in the NAM5 section of chapter 7.

If you installed an upgrade to CDCNET, follow the instructions in Activating a New Level of CDCNET in the CDCNET section of chapter 7.

If you did not build a new deadstart tape to install your component order, your installation is complete. If you wrote a new tape, continue to Step 7: Deadstart the New System.

Step 7: Deadstart the

Deadstart the system using the new deadstart tape. When the system comes up, it will be using all of the newly released software and permanent files. The new system is ready for production.

Your component order installation is now complete.

SOLVER Modification Sets	5-1
Using Modify for NOS and NOS Products	5-1
Adding Corrective Code	5-2
Creating a New Permanent OPL	5-2
Putting Code on a USER File	5-2
Using Update for Common Product Set, Network Host Products, CCP, and CROSS	5-3
Adding Corrective Code	5-3
Creating a Permanent NEWPL	5-4
Putting Code on a USER File	5-4
CDCNET Batch Critical Update (BCU)	5-5
Preliminaries	5-5
Procedure	5-5
Cleanup	5-7
Dual State Batch Critical Update (BCU)	5-8
Correcting the Dual State Partner	5-8
Correcting an Earlier Version of NOS	5-9
Correcting an Older Version of NOS/VE	5-10

This chapter describes how to apply a critical modification set from SOLVER, a CDCNET Batch Critical Update (BCU), and a Dual State Batch Critical Update (BCU).

SOLVER Modification Sets

Applying corrective code requires you to rebuild the software you are correcting. For general information about building products from source code, refer to chapter 6. These instructions may be executed from an interactive terminal logged in to your installation user name.

Using Modify for NOS and NOS Products

When corrective code is written against NOS or the NOS products, it is written in standard Modify format. A Modify PSR code modset consists of the following:

- **An Ident Card.** The ident card consists of the modset identifier (for a modset against a single deck, the identifier is the deck name followed by a sequence number. For a modset against more than one deck, the identifier is NS2 followed by a sequence number), the author's initials, and the modset creation date. Here is an example:

```
*IDENT      1DS13  JDN.          85/11/13.
```

- **Comment Lines.** A minimum of five comment lines are included with each modset, and these comment lines include the following information: the operating system identifier (PSR level), the PSR number to which the modset applies, code dependency information, the problem description, and the solution description. For example:

```
*/          ****   NOS 2.4.2-642 OS.
*/          ****   NS23316.
*/          ****   REQUIRES - NS2393(NS20749) .
*/          ****   PROBLEM - UNABLE TO *IDLE* A SUBSYSTEM
*/          IF IT IS ROLLED OUT BECAUSE THE JSN IS NOT BEING
*/          SAVED PRIOR TO THE *CEFM* FUNCTION.
*/
*/          SOLUTION - SET THE JSN PRIOR TO CALLING *CEFM*
```

In this example, the operating system identifier indicates the modset was written against a NOS level 642 system. This means the modset will install correctly on a level 642 system. The modset may or may not work on an earlier system and will probably be included in any system newer than 642. The REQUIRES statement indicates that you should also pick up modset NS2393 from SOLVER (PSR NS20749) since this modset depends on that earlier modset.

- **Corrective Code.** The corrective code to fix the problem consists of one or more DECK directives (specifying the name of the deck to be modified) followed by the assembler or compiler statements for that deck. For example:

```
*DECK      1DS
*I,NS2262.34      (2955)
                LDD      CM      SET JSN
                STD      CM+3
```

- **Comment Line.** A final comment line indicating the end of the modset.

```
*/          END OF MODSET.
```

Adding Corrective Code

There are two methods you can use to add corrective code to NOS and the NOS products:

1. Apply the Modify modset to OPLpsrin by using the MODOPL procedure on file INSTALL and then run the appropriate installation procedures.
2. Place the corrective code on a USER file and run the appropriate installation procedures.

The following examples illustrate the two methods.

Creating a New Permanent OPL

The example below illustrates how to install a sample modset named 1DS13. First, use the MODOPL procedure to apply the modset to OPLpsrin. This is done by using the USERF parameter.

```
GET, 1DS13 .
BEGIN, MODOPL, INSTALL, USERF=1DS13 .
```

MODOPL creates a new program library named OPLpsrout that includes the change. Note that the USERF file may contain multiple modsets, each separated by an end-of-record mark.

Next, run the affected installation procedure to generate new binaries. For example:

```
BEGIN, NOS, INSTALL .
```

Putting Code on a USER File

The USER file option allows you to add corrective code to a product without installing the code on the output program library. The code in the USER file is added only to the generated binaries.

For example, to build NOS with PSR 1DS13, first put the code on a permanent file (in this example, file 1DS13) and use the USERF parameter on the call to the NOS installation procedure:

```
BEGIN, NOS, INSTALL, USERF=1DS13 .
```

Using Update for Common Product Set, Network Host Products, CCP, and CROSS

When corrective code is written against the Common Product Set, Network Host Products, CCP, or CROSS, it is written in a standard Update format. An Update PSR code modset consists of the following:

- An Ident Card. The Ident card consists of the product name and, if there is a dependency with any other ident, a K parameter. For example:

```
*ID CPS2658,K=CPS056
```

The K=CPS056 indicates a code dependency. That is, CPS056 must either exist on the program library or precede CPS2658 in the Update input for this Ident to be processed.

- History Text. The history text begins with a *B HISTORY.2 record. The HISTORY comments are added to the HISTORY deck, which includes a description of the problem, the solution, any dependencies, the author's name, the date, decks affected by the product, and a *C HISTORY record. For example:

```
*B HISTORY.2
CPS2658      COMPASS/SCD.  A NUMERIC DATA ITEM WITH TWO PERIODS CAUSES
              COMPASS TO HANG.  CODE MODIFIES *NDS* IN *SCD* TO UPDATE
              THE COLUMN POINTER BEFORE TAKING ERROR EXIT.
              DEPENDENCY=CPS056
              AM      85/11/13      COMPASS
*C HISTORY
```

- Corrective Code. The corrective code to fix the problem consists of assembler or compiler statements:

```
*D CPS056.1,CPS056.2      (NR COMPASS.3328)
ZR  B2,NDS2A      IF FIRST *.*
SX6 A1-CARD+1     ERROR, BUT FIRST SET CARD POINTERS
SA6 COLUMN
BX7 X1
SA7 CHAR
EQ  ERR

NDS2A      SB2  B1      SET REAL NUMBER IN FLAG
*C COMPASS
```

- Comment Line. The comment line indicates the total number of cards in the modset.

```
*/ THERE ARE      20 CORRECTION CARDS INCLUDING THIS COMMENT.
```

Adding Corrective Code

There are two methods you can use to add corrective code to a product:

1. Use Update to create a NEWPL and then build the product.
2. Place the code on a USER file and build the product.

The following examples illustrate each of the methods.

Creating a Permanent NEWPL

This example illustrates how to use an Update modset, CPS2658, to fix a COMPASS problem. First, use an UPDATE command to add this code file and create a NEWPL:

```
ATTACH, OLDPL=CPS1psrout.
GET, CPS2658.
UPDATE, F, N, I=CPS2658, C=0.
```

Make the NEWPL permanent and run the installation procedure for COMPASS. In addition, you may want to back up your release version of the program library:

```
CHANGE, OCPS1PL=CPS1psrout.
DEFINE, CPS1psrout.
COPYEI, NEWPL, CPS1psrout.
RETURN, CPS1psrout.
BEGIN, COMPASS, INSTALL.
```

Putting Code on a USER File

The USER file option allows you to add corrective code to a product without creating a permanent NEWPL. The code is applied to the input PL and a temporary NEWPL, called NEWER, is generated. All assemblies and compilations are done using the compile file from NEWER.

- **Product Set and Networks.** All the Common Product Set and the Network Host Product installation procedures give you the option of specifying a file for input code via the USERF parameter.

For example, to build COMPASS with Ident CPS2658, first put the code on a permanent file (in this example, file CPS2658) and use the USERF parameter on the call to the COMPASS installation procedure:

```
BEGIN, COMPASS, INSTALL, USERF=CPS2658.
```

- **CCP and CROSS.** To add user code to CCP, the code must be on the permanent file UCCP for the CCPH1, CCPBLB, and variant installation procedures. For the CROSS installation procedure, put the code on file UCRS. The installation procedures for these products automatically look for these file names. That is, you do not need to specify the files on the call to the installation procedures. For more information, refer to CCP/CROSS Permanent Files in the CCP section of chapter 7.

CDCNET Batch Critical Update (BCU)

There are three potential types of corrections for CDCNET: a SOLVER correction to CHA1, a BCU to DCNS, and a problem that requires both a BCU and a SOLVER correction. If you are just installing a SOLVER correction to CHA1, follow the instructions earlier in this chapter for installing a correction to an Update product. The following instructions describe how to install a BCU alone and a BCU that requires a SOLVER correction.

Control Data will send you a BCU tape when you require a correction to a critical problem with CDCNET at your site. This correction will be to the DCNS portion of CDCNET. The RECLAIM-formatted tape you receive will contain the file PFGBCU1 which is installed by the procedure BCU in DCNPLIB. The installation of a BCU takes place on the network administration user name and should be executed from an interactive terminal.

NOTE

If you have changed the default Control Data installation user names (INSTALL, NETOPS, and NETADMN) to ones that are site-defined, the BCU installation process installs the files to the new user names. That is, a BCU installation retains the current user names; you cannot change these user names via the BCU installation process.

- The following backlevel support situation is not supported:

You have upgraded to use new user names and need to go back to a level of CDCNET that does not support the changing user name capability.

- The following backlevel support situation is supported:

You have upgraded to a level of CDCNET that supports the changing user name capability, you are using the Control Data default user names, and you need to go back to a level of CDCNET that does not support this capability.

Preliminaries

Before you begin these instructions, have the BCU tape available to be mounted and note the tape VSN and its density. (The VSN is listed in the media number field of the external tape label.)

Procedure

1. Log in to IAF under the network administration user name.
2. Install the BCU by entering the following procedure call:

```
BEGIN,BCU,DCNPLIB,vsn,density.
```

Replace vsn with the VSN of the BCU tape. Replace density with the tape density of the BCU tape (HY, HD, PE, or GE).

As this procedure executes, your terminal displays a message indicating the CDCNET version level being installed. It also displays NETFM output while the new boot files and object library are added to the directory. Check the NETFM output to ensure that all CREATE directives are completed normally. When the procedure is completed, all new CDCNET software files are loaded to the network administration user name. Site-controlled configuration files, such as the exception list, are not loaded.

3. If you are installing a SOLVER correction with a BCU, follow the directions given earlier in this chapter for installing a correction to an Update product. Be sure you use a new ID=id parameter when you invoke the CHA1 installation procedure. If the SOLVER modset changes any of the network utilities or servers that reside on the deadstart tape, you must generate a new deadstart tape.
4. If a SOLVER correction was applied, you must use the new NETITF binaries in file GLOBLIB. Access these binaries by entering the following commands:

```
ATTACH, GLOBLIB/UN=insun.
LIBRARY, GLOBLIB/A.
```

5. Generate a new combined message template file using the GENMSG procedure in DCNPLIB. Use the following procedure call and execute the procedure from the network administration user name:

```
BEGIN, GENMSG, DCNPLIB, ID=id, P=psrout[, PURGE].
```

Parameter	Description
id	The ID letter associated with the CHA1 template file name, which has the format HTF_id_psrout.
psrout	The NOS PSR level associated with the CHA1 template file name, which has the format HTF_id_psrout.
PURGE	This parameter is optional and can be specified to delete an existing combined template file. If you do not provide the PURGE parameter, GENMSG does not delete the file.

The GENMSG procedure produces a combined template file with file name TF_id_vvvv. The DCNS version level vvvv defaults to the version level of the BCU just installed, which causes the CDCNET template file, CTF_vvvv, to be used. For more information, refer to the CDCNET section in chapter 7.

6. Activate the new software and the message template file.
 - a. Activate the new software by executing the SETVL (SET_VERSION_LEVEL) procedure in DCNPLIB. This procedure updates file INITDCN and the exception list to indicate the new default versions for all CDCNET software, namely, ANACD, MANCC, NPA, host-connected and non-host-connected DIs, and the message template file. Here is the format for the SETVL procedure call:

```
BEGIN, SETVL, DCNPLIB, ALL, V=vvvv, ID=id.
```

Parameter	Description
-----------	-------------

vvvv	The version level of the BCU just installed.
id	The identifying letter of the combined message template file.

For a complete discussion on the SETVL procedure and version levels, refer to Activating a New Level of CDCNET within the CDCNET section of chapter 7.

Once SETVL has completed, all future load requests for DIs and all accesses for ANACD, MANCC, and NPA will use vvvv level software. The next initiation of NETOU will use template file TF_id_vvvv.

- b. Put the new message template file into use by Network Operator Utility (NETOU) by following the steps below:
 - (1) Find the job sequence name (JSN) of NETOU on the K,NAM and K.ST displays (refer to the NOS Version 2 Analysis Handbook).

- (2) Enter the following command:

```
DROP, jsn.
```

Replace jsn with the job sequence name of NETOU.

- (3) Restart NETOU by entering the following command:

```
X.NAMI (RS=OU)
```

7. If a SOLVER modset was installed that corrected a problem in the CHA1 network servers or utilities, deadstart the system using the new deadstart tape created in step 3.

The installation of the critical correction is now complete.

Cleanup

Once the new CDCNET version is running satisfactorily, you can remove earlier versions of the CDCNET software from disk to conserve space. This is done using the procedure CLEANUP, which purges software files from the network administration user name. Only files from the specified version level are removed. Boot files and object libraries are also removed from the network directory file, NETDIR. Execute the following procedure call from the network administration user name to specify the version level to be removed:

```
BEGIN, CLEANUP, DCNPLIB, V=vvvv.
```

Replace vvvv with the version level you want to remove from disk.

Dual State Batch Critical Update (BCU)

A BCU for a dual state system corrects a critical problem in the dual state routines that reside on the NOS deadstart tape. The BCU is on a permanent file tape. This tape contains binaries for the deadstart tape and a permanent file. The permanent file (PFGBCU2) is a multifile file that has the same format as the permanent file for DUAL (PFGDUAL). File 1 contains the Dual State Source Library, file 2 contains the VEMEM procedures, and file 3 contains a RECLAIM dump of NOS/VE binary support for back levels of NOS.

There are three options for installing a dual state BCU:

1. Install the corrective code onto a NOS system that was released with your version of NOS/VE. This version of dual state is the official NOS/VE partner to NOS.
2. Install the corrective code onto a version of NOS that was released prior to the version of NOS/VE you are using.
3. Install the corrective code onto a NOS system that is running an older version of NOS/VE. For example, NOS is running at L688 and NOS/VE is running at L678.

NOTE

The official NOS/VE partner, and the back levels of NOS that are supported, are shown in the NOS/VE Installation Handbook.

Correcting the Dual State Partner

If the NOS version you are running was released with the NOS/VE version you are running, execute the following steps. However, before you begin, you should have the BCU tape available for mounting and note the density and VSN of the tape. (The VSN is listed in the media number field of the external tape label.) Also, you should have an unlabeled scratch tape available. This tape will be used for a new deadstart tape and will be requested with a VSN of NDT.

Execute these commands from an interactive terminal logged in to the installation user name.

1. Access the current running system by entering this command:

```
COMMON, SYSTEM.
```

2. Execute SYSGEN:

```
SYSGEN, UPGRADE, SYSTEM, NEW, vsn, density1, density2.
```

Replace *vsn* with the VSN of the BCU tape. Replace *density₁* with the tape density of the BCU tape. Replace *density₂* with the tape density of the new deadstart tape.

The procedure first requests the BCU tape. After the BCU tape is processed, the procedure requests the unlabeled scratch tape and writes the new system to this tape.

Correcting an Earlier Version of NOS

If the NOS version you are running was released prior to the NOS/VE version that you are running, execute the following steps. Before you begin, you should have the BCU tape available for mounting and note the density and the VSN of the tape. (The VSN is listed in the media field of the external tape label.) Also, you should have an unlabeled scratch tape available. This tape will be used for a new deadstart tape and will be requested with a VSN of NDT.

1. Log in to your installation user name using an interactive terminal.

To properly install a BCU, you need the SYSGEN procedures that match the PSR level of NOS/VE. This means you need the procedure SYSGEN and the user library PFGLIB available. If you do not have these on your catalog, they may be GTRd from the released NOS deadstart tape. The local file names must be SYSGEN and PFG, respectively.

The level of software in the RECLDB file must also match the PSR level of NOS/VE. You can check this by cataloging the RECLDB file. Record 1 shows the user name known to RECLAIM (for example, NS2psrin). If the user name does not contain the correct PSR level, you must change the name of the RECLDB file to something else. After installing the BCU, delete the newly created RECLDB and change the old name back to RECLDB.

2. Enter this command:

```
SYSGEN, UPGRADE, 0, 0, vsn, density.
```

Replace vsn with the VSN of the CDC-supplied BCU permanent file tape. Replace density with the tape density of the BCU permanent file tape (HY, HD, PE, or GE).

This command updates the RECLAIM database to contain information about all of the files on the BCU permanent file tape.

3. To obtain the corrected dual state binaries from the BCU tape, enter the following command:

```
SYSGEN, DUAL, OLDNOS, BCU.
```

This command creates or replaces files with the naming convention NVEoldlev, where oldlev is the PSR level of NOS for which the binaries were compiled.

4. To apply these binaries to your production system and create a new deadstart tape, use the following commands:

```
COMMON, SYSTEM.  
ATTACH, NVEoldlev.  
SYSGEN, DST, SYSTEM, NVEoldlev, NEW, 0, density.
```

Replace oldlev with the NOS level you are running and replace density with the tape density of the new deadstart tape.

Correcting an Older Version of NOS/VE

In this situation, you are running a new NOS system with an older version of NOS/VE. For example, NOS is running at L688 and NOS/VE is running at L678. You have received a BCU tape for the level of NOS/VE you are running.

To install this BCU, you need to build new binaries for the dual state combination you are running from the source available on the BCU tape.

Follow the instructions in Dual State Upgrade of NOS Only in the DUAL section of chapter 7.

Introduction	6-1
Customizing Deadstart Tape Binaries	6-2
Step 1: Load the Installation Tools and Files	6-2
Step 2: Customize the Installation Procedures	6-3
Step 3: Create USER Files	6-4
USERF Parameter	6-5
Step 4: Create Files GLOBLIB and PRODUCT	6-6
Step 5: Execute the MODOPL Procedure	6-8
Step 6: Build Products from Source Code	6-9
Run Group 1	6-12
Run Group 2	6-12
Run Group 3	6-12
Run Group 4	6-12
Run Group 5	6-12
Run Group 6	6-12
Step 7: Create a New Deadstart Tape	6-22
Customizing Permanent Files	6-23

Introduction

This chapter describes how to customize the binaries for products that reside on the deadstart tape as well as how to customize the released permanent files. Note that customization is performed as part of one of the installations described in chapters 2, 3, 4, and 5 of this handbook. This chapter contains the following sections:

- Customizing Deadstart Tape Binaries
- Customizing Permanent Files

Before you begin the steps in this chapter, note the following:

- Make a list of the products you want to customize. Next, check the product dependency charts (figures 6-1 through 6-3) for other products that are dependent on the products you want to customize and add these to the list. These are the products you must rebuild.
- Refer to chapter 7 for special information about each product you are installing.
- Special installation parameters, procedures, and commands relating to DECKOPL are described in chapter 9.
- Instructions for loading individual files (for example, the PSR report or DECKOPL) are described in chapter 8.
- If you are doing a component installation and your order contains a Modify product that resides on the composite OPL, you must update your existing composite OPL before customizing any products. Use the SYSGEN(RECLAIM) command to load file OPLpsrin from the component order permanent file tapes. This file must be LIBEDITed against your previous composite OPL.
- The *R1G* register should not be altered by the site while customizing binaries. If the UPROC on your installation user name sets the *R1G* register to a nonzero value, you should reset *R1G* to zero before you begin customizing binaries.
- Special information about installing a 63-character set system is detailed in appendix C. Read this appendix before you begin installing your system.
- 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 load the source installation tools, you can set up files for disk installation. Refer to Disk Installations in chapter 9 for information about controlling the permanent file environment.
- If you wish to run a newer version of NOS with an older version of NOS/VE (for example, NOS L688 with NOS/VE L678), the dual state binaries must be reassembled. Refer to Dual State Upgrade of NOS Only in the DUAL section of chapter 7. If other products need to be customized during the upgrade to the new NOS level, you should

assemble them first by following the instructions described in this chapter. This ensures that the correct output PLs are available when customizing dual state.

- If all network-related products are ordered, the installation user name may require unlimited indirect access file validations because the size of NAMSTRT will exceed the default indirect access file size.

Customizing Deadstart Tape Binaries

Step 1: Load the Installation Tools and Files

If you are customizing as part of a first-time, component, or corrective installation, only perform step 1 below. If you are customizing as part of an upgrade installation, only perform step 2 below.

1. Have the released permanent file tapes available for mounting and execute the following command from an interactive terminal logged in to the installation user name:

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

Include the keyword CCP only if you want to load CCP/CROSS binary installation files. Refer to chapter 7 for more information about CCP installation.

NOTE

If the source build procedures for the current level of NOS are already on the installation user name, skip this step.

2. Have the released permanent file tapes available for mounting and execute the following command from an interactive terminal logged in to the installation user name:

```
PURGE, RECLDB/NA.  
GTR, SYSTEM, PFG.ULIB/PFGLIB  
BEGIN, SYSGEN, SYSTEM, SOURCE, CCP.
```

Include the keyword CCP only if you want to load CCP/CROSS binary installation files. Refer to chapter 7 for more information about CCP installation.

If you will be running the SEED procedure in the same terminal session (see Step 4), do not RETURN file SYSTEM. Keep file SYSTEM as a local file to avoid an additional tape request in Step 4.

The procedures in steps 1 or 2 above load the RECLAIM database (RECLDB), DECKOPL, INSTALL, CODEPL, MISCPL (if released), NOS PSR report file (PSRRPT), USERBPS, NAMSTRT (if appropriate), and COMMOD and make RECLAIM a permanent file. Refer to chapter 8 for a complete list of files.

Step 2: Customize 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 system, read appendix C before continuing your installation.

1. You may want to print a DECKOPL listing by using the DECKLIS procedure. (This listing is over 400 pages.) For information about additional DECKLIS parameters, refer to chapter 9.

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

2. Edit file COMMOD using an available editor. File COMMOD is a correction set containing parameters that control the following:
 - Whether installation is from tape or disk.
 - Where files reside for a disk installation.
 - USER and CHARGE commands used for executing installation procedures.
 - Whether USER files are searched for automatically.
 - Different values for other common parameters. For example, parameters that specify what job output listings to create, parameters that determine whether job output should be printed, parameters that specify the tape density for tape requests, parameters that determine whether output PLs are written, and a parameter that determines the type of load map that is generated.

Refer to COMMOD File Parameters in chapter 9 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 finish, replace file COMMOD.
4. When customizing some products (for example, CHA1), files are created on the installation user name. In general, these files are private; however, some of them are used during the installation process. The PERMIT procedure in DECKOPL permits these files to the appropriate user names. If you change the default Control Data installation user names (INSTALL, NETOPS, and NETADMN), PERMIT needs to be updated to reflect the site's user names. For example, PERMIT uses the following IF test to permit NLFFILE:

```
.IF, $SYMFN$ .EQ. $NLFFILE$, PERMIT3 .
  PERMIT (REALFN, NETOPS=R)
.ENDIF, PERMIT3 .
```

When NLFFILE is created, the following statement calls the PERMIT procedure:

```
BEGIN (PERMIT, INSTALL, REALFN=G_GN_CL, SYMFN=NLFFILE)
```

A site can add a PERMIT call to any installation procedure that creates permanent files. A matching IF test must be added to the PERMIT procedure. These changes should be added to file COMMOD so that when SETUP is run in the following step, the INSTALL file is created with the desired changes.

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

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

Step 3: Create USER Files

To further customize products during installation, you can create USER files (files containing modified code for each product you want to customize). (User code for the composite OPL is discussed in Step 5.) Each file can contain the following:

- Programming Systems Report (PSR) corrective code from SOLVER. Refer to chapter 5 for information about SOLVER code.
- MISCPL code (refer to the MISCGET procedure in chapter 9).
- Product installation parameter settings (for specific parameters for a product, refer to that product entry in chapter 7).
- Site code.

Refer to chapter 7 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, PFTF, 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 procedure, you can use the USERF parameter. The USERF parameter should be controlled from COMMOD (refer to Step 2: Customize the Installation Procedures in this chapter and COMMOD File Parameters in chapter 9).

- If USERF is null (the default value), a USER file is looked for only if a file name is specified on the call to the installation procedure. For example:

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

In this case, the installation procedure looks for a permanent file named USERAAM. If it does not find the file, the procedure aborts. If USERF=UJOBNAM is specified on the procedure call, it overrides the null value.

- If USERF is equal to UJOBNAM, a USER file is looked for automatically. If a permanent file with the name U concatenated with the first six characters of the installation procedure name is found, the user code is applied. If no permanent file of that name is found, the procedure executes as usual and no user code is applied. For example:

```
BEGIN, AAM2, INSTALL.           (looks for a file named UAAM2)
```

```
BEGIN, BINEDIT, INSTALL.       (looks for a file named UBINEDI)
```

If USER=filename is specified, it overrides the UJOBNAM value.

NOTE

If you create a USER file to add site modifications, the code is not installed on the output program library. This code is applied last and is added only 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, manually create a new program library that contains the user code using Modify or Update.

Step 4: Create 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.

The SEED procedure does the following:

- Places necessary user libraries on file PRODUCT.
- Places the following types of binaries on GLOBLIB:
 - Compilation, assembly, and loader tools
 - System text definition binaries
 - Configuration tools

To execute the SEED procedure, you must have a copy of the released system:

- If you are performing a first-time, component, or corrective installation, specify DST=SYSTEM on the SEED call to use the running system for initializing PRODUCT and GLOBLIB.
- If you are performing an upgrade installation, specify DST=SYSTEM on the SEED call to use the local file SYSTEM created in Step 2 of chapter 3, Upgrade Installation. If file SYSTEM is not local, REQUEST the released deadstart tape and COPY it to a file named SYSTEM. Then specify DST=SYSTEM on the SEED call.

NOTE

The SEED procedure assumes the Maintenance Package, which contains SYMPL, was ordered. Without SYMPL, the SEED procedure will abort. If you want to rebuild a product that does not require SYMPL and you did not order the Maintenance Package, execute the NOEXIT command before running the SEED procedure. After SEED is completed, type in ONEXIT.

Here is the format for the SEED call:

```
BEGIN, SEED, INSTALL, param1, . . . , paramn
```

Parameter	Description
DST=dstname	Local file name for your copy of the system. If file dstname is not a mass storage file, dstname is copied to a mass storage file before the GTR commands in the SEED procedure extract the required binaries. If DST=SYSTEM and no file named SYSTEM is local or on disk, SEED performs a COMMON(SYSTEM) command for you.
NOREBLD	The keyword parameter NOREBLD causes SEED to load only a minimal set of binaries into GLOBLIB and PRODUCT. Do <i>NOT</i> specify NOREBLD unless you plan to build ALL products, for example, if converting to a 63-character set system.
RESET	The keyword parameter RESET causes SEED to initialize the files JOBSTAT and DAYFILS; these files keep statistics on the installation procedures and dayfiles. To get a report of the resource usage of your installation procedures and a list of what procedures have passed or failed, refer to the REPORT procedure in chapter 9.

Step 5: Execute the MODOPL Procedure

Use the MODOPL procedure to apply code (for example, to modify installation parameters) against the composite output program library (OPL). The OPL contains all Modify-formatted products except APL 2 and PFTF.

If you do not want to write a new OPL – that is, you are not applying any modifications to the composite OPL and are not changing the value of psrout (see COMMOD File Parameters in chapter 9) – you do not need to execute MODOPL. The OPL will be automatically loaded from tape when it is needed.

If you change psrout or will be applying code, you must execute MODOPL. If you execute MODOPL, it creates disk file OPLpsrout (psrout defaults to the current release level). 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 COMXOPL in DECKOPL defines its residency. OPLpsrout is used as an auxiliary PL by many installation procedures. The operating system installation procedures use it to generate the compile file for operating system installations.

To apply code against the composite OPL, create a USER file. This file may include any of the following items:

- Miscellaneous code (optional) from MISCPL (refer to the MISCGET procedure in chapter 9).
- Modifications of the NOS installation parameters (refer to chapter 7).
- Modifications to operating system products (refer to chapter 7).
- Site code.

When MODOPL executes, 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 procedure, you can use the USERF parameter. The USERF parameter should be controlled from COMMOD (refer to Step 2: Customize the Installation Procedures in this chapter and COMMOD File Parameters in chapter 9).

- If USERF is set to null (the default value) in file COMMOD, a USER file is looked for only if a file name is specified on the call to the installation procedure. For example:

```
BEGIN, MODOPL, INSTALL, USERF=NOSMODS.
```

In this case, MODOPL looks for a permanent file named NOSMODS. If it does not find the file, MODOPL aborts. If USERF=UJOBNAM is specified on the procedure call, it overrides the null value.

- If USERF is set to UJOBNAM in file COMMOD, a USER file named UMODOPL is looked for automatically. If file UMODOPL is found, the user code is applied. If no file of that name is found, MODOPL executes as usual and no user code is applied. For example:

```
BEGIN, MODOPL, INSTALL. (looks for a file named UMODOPL)
```

If USERF=filename is specified, it overrides the UJOBNAM value.

Note that you can only permanently apply the code on file USER (or the file specified with the USERF parameter) to the composite OPL by using the MODOPL procedure. For all other installation procedures, USER code affects only the binaries and not the new OPL.

If you are only changing the value of psrout and are not applying code, you can execute MODOPL by using this command:

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

Step 6: Build Products from Source Code

To build products from source, you need to determine the related installation procedures, the dependencies they may have, and if there are any unique parameters. The following pages contain this information.

Refer to table 6-1 to make a list of associated installation procedures for the products you plan to modify.

NOTE

If you are changing the default Control Data installation user names (INSTALL, NETOPS, and NETADMN) to ones that are site-defined, note the following:

- You must rebuild SYSGEN and provide the site user names. For example:

```
BEGIN, SYSGEN, INSTALL, insun, netopun, netadun.
```

This changes PFGLIB so that files are permitted to the correct user names during execution of SYSGEN. Since SYSGEN checks the installation user name first for PFGLIB, the correct one will be used.

- There are 3 installation parameters in CHA1 that reference user names NETOPS and NETADMN (refer to Installation Parameters in the CDCNET section of chapter 7). It is recommended that these parameters be changed to match the user names desired by the site and that CHA1 be reassembled.
-

Table 6-1. OIP Options and Associated Installation Procedures

Product Name	Related Installation Procedures
APL 2	APL2
Automated Tape Facility (ATF)	ATF
BASIC 3	BASIC3
Cartridge Tape Subsystem (CTS)	CTS
COBOL 5	COBOL5/COBOL5Q, FCL1, FCL2, PMD
Communications Control Program (CCP) 3	CCPPH1, CCPBLB, CCPVAR, CCPLOAD
Concurrent Maintenance Library (CML)	No installation procedure. Consult the Concurrent Maintenance Library Reference Manual.
Control Data Distributed Communication Network (CDCNET)	CHA1
Conversion Aids Subsystem 3	LCS3, FCS3
CYBER CROSS System 1	CROSS
CYBER Database Control System 2	CDCS2
CYBER Interactive Debug	CID
Data Catalog 2 Version 2.0	DCL
Data Description Language	DDL3
Disk Array Subsystem (DAS)	DAS
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
Interactive Facility 1	IAF
Interactive Transfer Facility	ITF, RHC
Link Interface Program under CCP 3	(Part of CCP 3)
Maintenance Package	TOOLS, FORMAT, SYMPL
Mass Storage Extended	MSE
Message Control System 1	MCS

(Continued on next page)

Table 6-1. OIP Options and Associated Installation Procedures

Product Name	Related Installation Procedures
<i>(Continued from previous page)</i>	
MMCL V3, MAP Macro Control Library	MMCL
MSSI V3, MAP III	MSSI, AP1I, AP1L
Multi-Mainframe Module 1	MMF
Network Access Method 1	NAM5/NAM5D
NOS Online Manuals	No installation procedure.
NOS Scope 2 Station	NSS
NOS 2 Package	AAM2, BAM, BINEDIT, BIT8, CCL, COMPASS, FORM, LOADER, NIP5870, NOS, NOS2B, OSLIB, OSTEXT, PFTF, RDFEX, SYSGEN, TDU, TERMLIB, TEXT, TEXTIO, UPDATE
PASCAL 170	PASCAL
Printer Support Utility	PSU
PTF/QTF File Transfer Facilities	RHP, RHC
Query/Update 3	QU3
Remote Batch Facility 1	RBF5/RBF5D
Remote Host Facility	RHF, RHC
Sort/Merge 5	SORT5
Tape Management Subsystem (TMS)	TMS
TCP/IP/FTP/TELNET	TCPH
Tracer 1	TRACER
Transaction Facility 1	TAF
Xedit 3	XEDIT
3270 TIP for CCP	(Part of CCP 3)

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. Figures 6-1, 6-2, and 6-3 illustrate the installation procedure dependencies.

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

NOTE

If a product has two possible installation procedures, the procedures for the product are enclosed in one box in the Build Dependencies Charts (figures 6-1, 6-2, and 6-3). Run only one procedure for each product.

Run Group 1

Run the group 1 procedures listed in table 6-2 in the order in which they are shown in figure 6-1. Do not run a procedure until the preceding procedure has finished.

Run Group 2

Run the group 2 procedures listed in table 6-3 after running all the procedures in group 1. You can run the group 2 procedures in any order you want and more than one procedure can be run simultaneously. (See figure 6-2.)

Run Group 3

Run the group 3 procedures listed in table 6-4 after all the procedures in group 1. Group 3 procedures must be run in the order shown in figure 6-2.

Run Group 4

Run the group 4 procedures listed in table 6-5 after all the procedures in group 1. Group 4 procedures must be run in the order shown in figure 6-3.

Run Group 5

Run the group 5 procedures listed in table 6-6 after all the procedures in group 1 and after the AAM2 procedure from group 3. Group 5 procedures must be run in the order shown in figure 6-3.

Run Group 6

Run the group 6 procedure listed in table 6-7 after the NAM5 procedure from group 5 and after the FTN5 procedure from group 4.

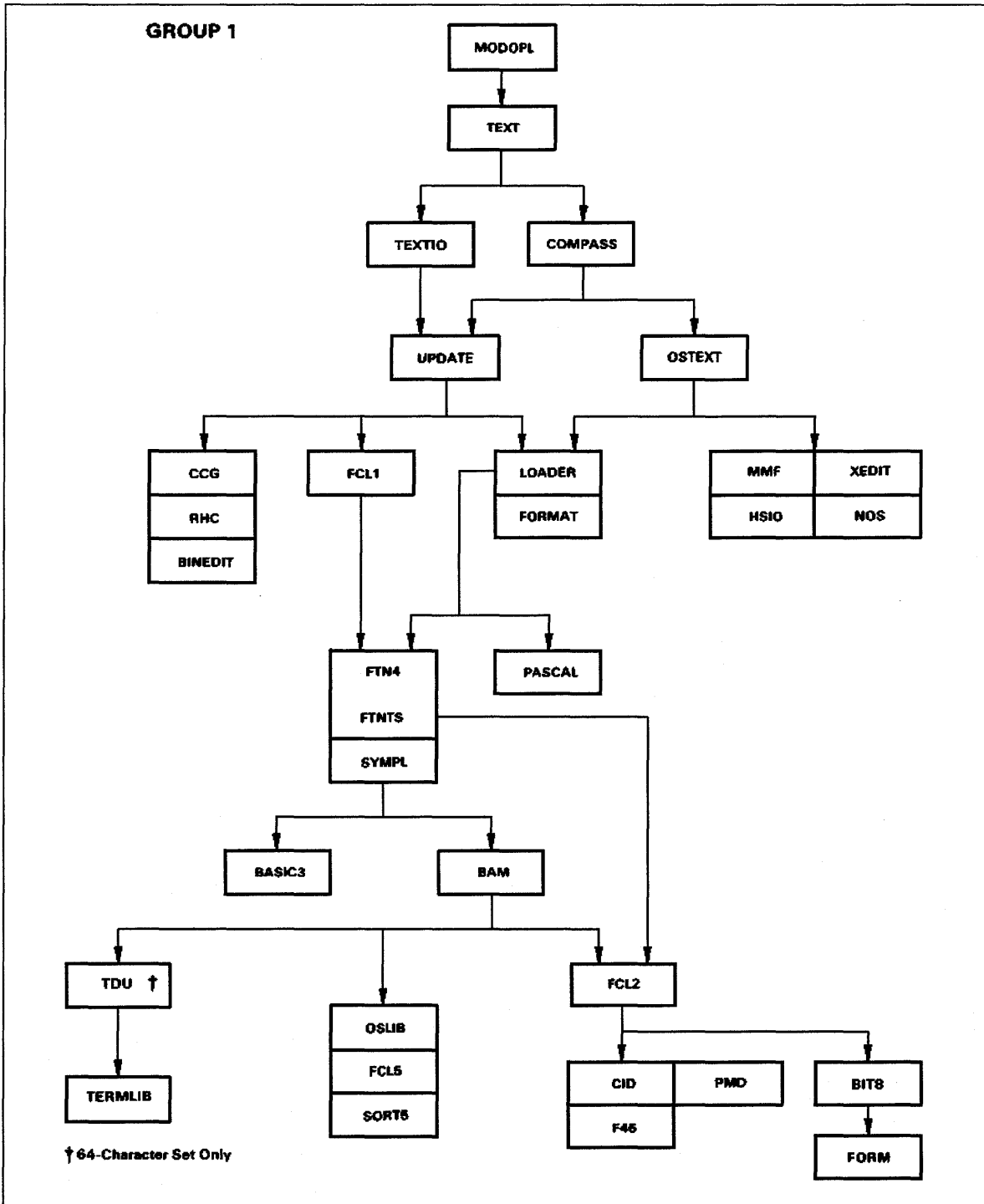


Figure 6-1. Build Dependencies Chart (Group 1)

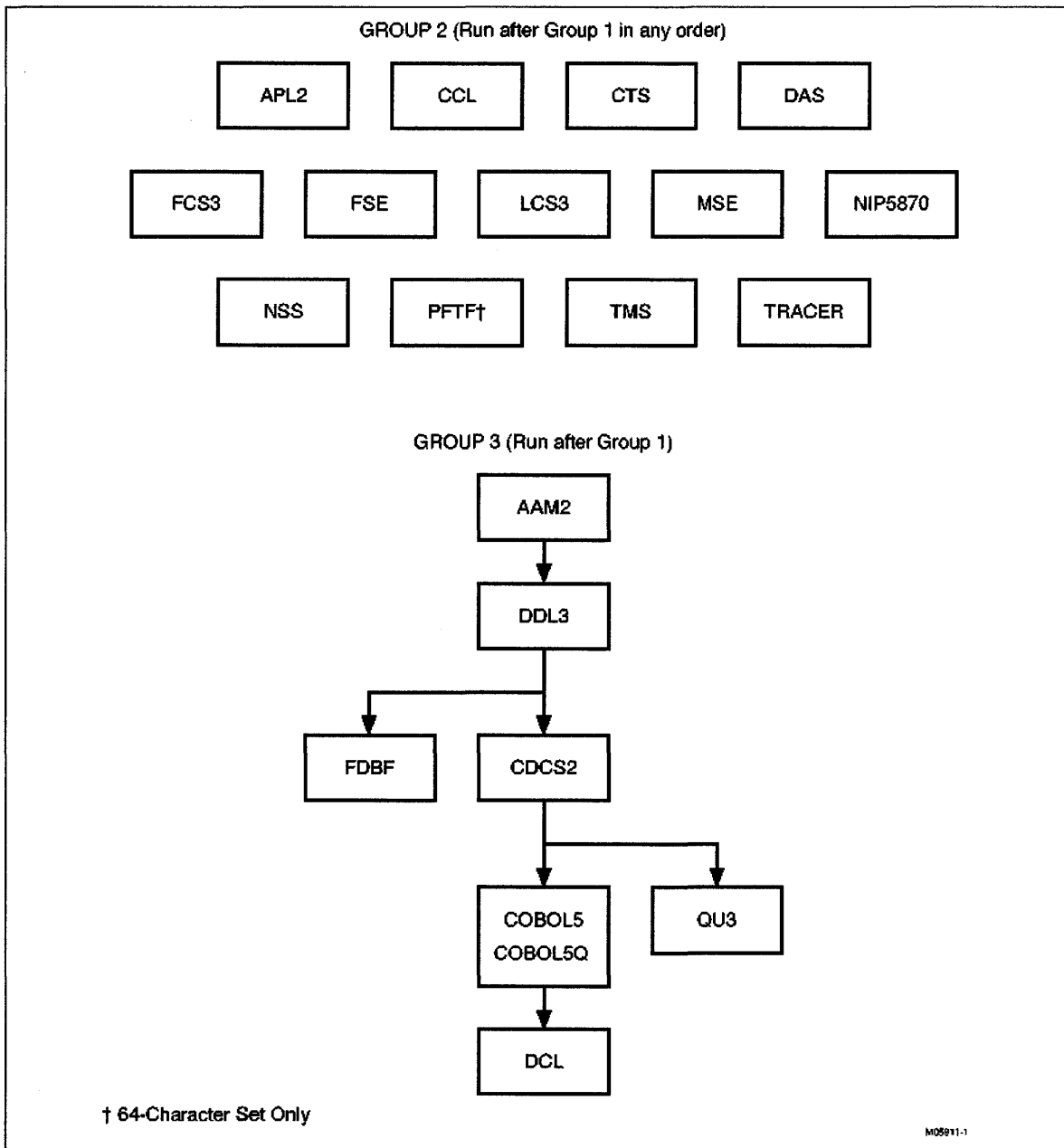


Figure 6-2. Build Dependencies Chart (Groups 2 and 3)

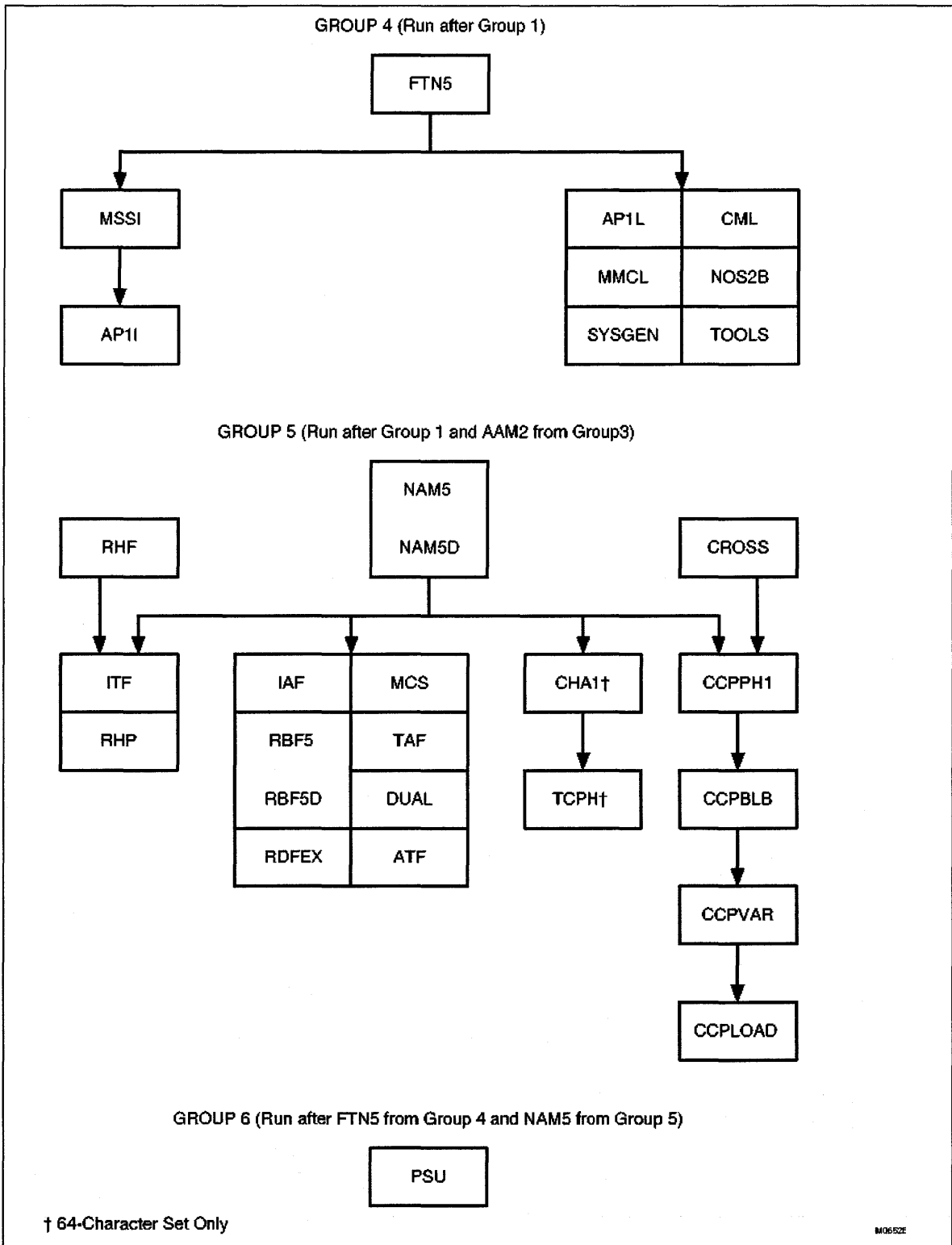


Figure 6-3. Build Dependencies Chart (Groups 4, 5, and 6)

If you modify a product that affects another product, you may need to add the other product's installation procedure to your list. Use the dependencies charts (figures 6-1, 6-2, and 6-3) to determine which installation procedures to run. If you are still unsure whether the modification you want to make will affect another product, refer to the installation procedures in DECKOPL.

The basic format for executing an installation procedure from the INSTALL file is the following:

```
BEGIN, procname, INSTALL, p1, . . . , pn.
```

Replace *procname* with the installation procedure name. The optional parameters (*p*₁ through *p*_{*n*}) include two types of parameters: common and unique.

Common parameters are explained in the COMMOD File Parameters section of chapter 9. Tables 6-2 through 6-7 list the unique parameters and required files for each installation procedure. Special information, such as unique parameters, is provided in chapter 7.

Table 6-2. Group 1 Products

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

(Continued on next page)

Table 6-2. Group 1 Products

Procedure Name	Unique Keywords	Required Files
<i>(Continued from previous page)</i>		
MMF		OPLpsrout
HSIO		OPLpsrout
XEDIT		OPLpsrout
NOS		OPLpsrout
FTN4		FTN4psrin CPS1psrout
FTNTS		FTI4psrin CPS1psrout
SYMPL		SYMPpsrin
PASCAL		PASCpsrin OPLpsrout
BASIC3		BAS3psrin CPS1psrout OPLpsrout
BAM		BAM1psrin TEXTpsrout
FCL5		FCL5psrin CPS1psrout
SORT5		SRT5psrin
OSLIB		OPLpsrout
FCL2		FCL4psrin CPS1psrout
BIT8		BIT8psrin
CID		CID1psrin OPLpsrout
PMD		PMD5psrin CPS1psrout
F45		F451psrin CPS1psrout
TDU ¹		TDU1psrin OPLpsrout
TERMLIB	TERMCAP=termcap	OPLpsrout
FORM		FORMpsrin
1. 64-character set only.		

Table 6-3. Group 2 Products

Procedure Name	Unique Keywords	Required Files
APL2	TERMTYP=termtyp	APL2psrin OPLpsrout
CCL		CCL1psrin CPS1psrout OPLpsrout
CTS		OPLpsrout
DAS		OPLpsrout
FCS3		FCS3psrin
FSE		OPLpsrout
LCS3		LCS3psrin
MSE	SAVELIB	OPLpsrout
NIP5870		OPLpsrout
NSS		NSS1psrin OPLpsrout
PFTF ¹		PFTFpsrin TDU1psrin OPLpsrout
TMS		OPLpsrout
TRACER		OPLpsrout

1. 64-character set only.

Table 6-4. Group 3 Products

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

Table 6-5. Group 4 Products

Procedure Name	Unique Keywords	Required Files
FTN5		FTN5psrin CPS1psrout CCG1psrout
MSSI	BLDMLIB, MSAMLIB	MSSIpsrin OPLpsrout
AP1I	MEMSIZE	AP1Ipsrin
MMCL		MMCLpsrin
AP1L	MAPTYP, ADD, MUL, DIV, SQRT	MMCLpsrin
TOOLS		OPLpsrout
NOS2B		OPLpsrout
SYSGEN	INSUN=insun, NETOPUN=netopun, NETADUN=netadun	
CML ¹		

1. Consult the Concurrent Maintenance Library Reference manual for installation information.

Table 6-6. 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, NOLOG, TASKLB	OPLpsrout
DUAL	TRACE, NOSLEV=noslev, VELEV=velev, CSET=cset	DUALpsrin OPLpsrout TDU1psrin BAM1psrout
ATF		ATF1psrin NAM5psrout OPLpsrout
CHA1 ¹	DEBUG, NOTRACE, ID=id	CHA1psrin NAM5psrout OPLpsrout

1. 64-character set only.

(Continued on next page)

Table 6-6. Group 5 Products

Procedure Name	Unique Keywords	Required Files
<i>(Continued from previous page)</i>		
TCPH ¹	DEBUG, TRACE	TCPHpsrin NAM5psrout OPLpsrout
CROSS		CRSSpsrin
CCPPH1	DIAG, REMT, BSTP, XTRAPLS	CCPBpsrin CCPTpsrin CCPRpsrin CCPDpsrin
CCPBLB	XREF	
CCPVAR	VN=vn	
CCPLOAD	GN=gn	
1. 64-character set only.		

Table 6-7. Group 6 Products

Procedure Name	Unique Keywords	Required Files
PSU	NOTRACE	PSU1psrin OPLpsrout

Step 7: Create a New Deadstart Tape

After you have completed building products customized for your site, create a new deadstart tape by entering this command from an interactive terminal logged in to the installation user name:

```
BEGIN,GENDST,INSTALL,SYSTEM=odt,NEW=ndt,D=density,LIST=list.
```

This command merges the binaries in file PRODUCT with the current system or another deadstart tape to produce a new deadstart tape. Refer to chapter 9 for more information about GENDST.

If you are building on a system with limited disk space, execute GENDST followed by RESETP to reduce the size of file PRODUCT. Refer to chapter 9 for information about the RESETP procedure.

If you are customizing a product as part of a component installation, use the deadstart file created in Step 2 of chapter 4 for the SYSTEM parameter. This tape contains the released binaries from the component order tape and ensures that the binaries are placed in the proper order.

NOTE

A deadstart file must fit onto a single reel of tape. If your deadstart file is on more than one tape, you will not be able to deadstart using the tapes. You should rebuild the tape using a higher tape density or a longer tape.

If your deadstart file still extends past the single reel limit, remove some binaries from the tape. Do not remove any of the first five libraries. Place the removed binaries on a permanent file and SYSEDIT this file after deadstart. You may want to put this SYSEDIT file into SYSPROC, which is always called after deadstart. Refer to the NOS Version 2 Analysis Handbook for information about SYSEDIT.

Customizing Permanent Files

In addition to the deadstart tape binaries, NOS uses permanent files to control the execution of subsystems, provide interactive help for commands, and provide information that describes the software and hardware configuration of peripheral equipment. This section describes how to customize the permanent files.

Modify the permanent files on the installation user name by using one of these methods:

- Apply user code during the build process (this is described in Step 3 and Step 5 of this chapter).
- Modify the permanent files after they are generated in the build process. The files can be modified using an available editor.
- Load the permanent files to disk from the permanent file tape and then modify the files. Chapter 8 describes how to load individual permanent files from the permanent file tapes using SYSGEN(RECLAIM). You can also call a SYSGEN function to load a file from the tape and break it into its component pieces. For example, SYSGEN(MSE1) loads file PFGMSE1 from the tape and creates files MSES LAV, MSE, and MSECONF.

Table 3-1 in chapter 3 (which shows files you can prevent from loading) and appendix B (which lists the names of all PFG files and their associated function names) can be used to determine permanent file names and SYSGEN function names.

Once the files have been modified, they must be moved to the proper user names on the system as described in chapter 3 or 4. To avoid overwriting current system files, do not move these files until directed to do so. For a first-time installation, refer to SYSGEN(MOVE) in chapter 8 for information about moving permanent files.

Special Product Installation Information 7

Subsystem Initiation	7-1
Subsystem Startup Files	7-1
Subsystem Enable and Disable Commands	7-1
Automatic Subsystem Initiation	7-1
AAM2 - CYBER Record Manager Advanced Access Methods Version 2	7-4
USER File Directives	7-4
Additional Procedures	7-4
APL2 - APL Version 2	7-7
Unique Parameters	7-7
APL Entry Message	7-7
SYSGEN Functions	7-7
APL Validation Requirements	7-8
ATF - Automated Tape Facility	7-9
Unique Parameters	7-9
ATF Command	7-9
File Placement	7-9
BAM - CYBER Record Manager Basic Access Methods Version 1	7-10
Installation Parameters	7-10
Installation Procedure Messages	7-10
BASIC3 - Basic Version 3	7-11
Installation Parameters	7-11
Installation Procedure Messages	7-11
CCL - CYBER Control Language Version 1	7-12
Installation Parameters	7-12
Additional Procedures	7-14
CCP - CYBER CROSS System and Communications Control Program	7-15
Hardware Requirements	7-16
Configuration Information	7-17
Released Software Variants	7-17
Build Steps Description	7-19
Binary Installation Option	7-23
General Build Step Call	7-24
CCP/CROSS Permanent Files	7-26
USERBPS File	7-29
Security Character Parameters	7-30
CROSS - CROSS System Installation	7-31
Installation Parameters for CROSS	7-32
CCPPH1 - CCP Phase 1	7-32
CCPBLB - CCP Binary Library	7-34
CCPVAR - CCP Variant	7-35
CCPEDIT - Edit Variant Module	7-36
CCPLOAD - Generate CCP Load File	7-37
CCPPURG - CCP/CROSS Installation Files Purge	7-38
CCP System Definition	7-38

CCP Variant Load Module Definition	7-40
CCP Load File Definition	7-42
CCP Extra PLs Definition	7-42
CCP/CROSS Installation Examples	7-43
CDCNET - Control Data Distributed Communications Network	7-55
Configuration Information	7-56
Installing CDCNET for the First Time	7-57
Upgrading CDCNET	7-59
Activating a New Level of CDCNET	7-62
Installing a CDCNET Separate TIP	7-62
Installing CDCNET for MDI Reset Support Only	7-63
Installation Wrapup Activities	7-64
NPA Database Maintenance	7-64
CDCNET Host Application (CHA1) Product Content	7-65
Unique Parameters	7-66
USER File Directives	7-66
Installation Parameters	7-66
Message Templates	7-68
NAM Startup Procedure File Changes	7-69
CDCNET Network Host Application Program Requirements	7-70
NETFS - Network File Server	7-71
NETLS - Network Log Server	7-72
NETOU - Network Operator Utility	7-73
INITMDI - Initialize MDI	7-74
Network Definition Language (NDL) Requirements	7-76
CDCS2 - CYBER Database Control System Version 2	7-77
Unique Parameters	7-77
CDC Procedure File	7-77
Special Notes	7-77
Accounting Table	7-77
CID - CYBER Interactive Debug Version 1	7-80
CML - Concurrent Maintenance Library	7-80
COBOL5 and COBOL5Q - COBOL Version 5	7-81
Installation Procedures	7-81
Unique Parameters	7-81
Installation Parameters	7-82
DCL - Data Catalogue 2 Version 2.0	7-83
Product Dependencies	7-83
SYSGEN Functions	7-83
DUAL - Dual State	7-84
Unique Parameters	7-84
Configuration Information	7-85
Dual State Source Library	7-86
Dual State Upgrade of NOS Only	7-87
Dual State on Older NOS Systems	7-90
NOS and NOS/VE Memory Allocation	7-90
Allocation of Dual State Peripheral Processors	7-91
Dual State Minimum Hardware Requirements	7-92

NOS/VE Initiation	7-93
VEIAF Family Name Selection	7-93
NOS SETVE Modification	7-94
NOS RUNJOBS Modification	7-95
NOS ACCFILE	7-96
NOS ACCFILE Modification	7-98
FDBF - FORTRAN Data Base Facility Version 1	7-99
Product Dependencies	7-99
Unique Parameters	7-99
FSE - Full Screen Editor	7-100
FSEEX and SMFEX Implementations	7-100
SMF Procedure File	7-101
Installation Parameters	7-102
Site-Defined Teach File	7-102
SYSGEN Functions for FSE	7-102
SMFSTAT Procedure	7-103
FTN4, FTNTS, FTN5 (FORTRAN Extended Version 4, FORTRAN Extended Version 4 with Interactive Option, FORTRAN 5)	7-104
MODEL Parameter	7-104
Installation Parameters for FTN4 and FTNTS	7-104
Installation Parameters for FTN5	7-104
IAF - Interactive Facility Version 1	7-105
Installation Parameters	7-105
IAF Procedure File	7-105
Special Notes	7-106
ITF - Interactive Transfer Facility Version 1	7-107
Product Dependencies	7-107
ITF Command	7-108
LCS3 and FCS3 - Conversion Aids System Version 3	7-109
USER File Directives for LCS3	7-109
LOADER - CYBER LOADER Version 1	7-111
Unique Parameters	7-111
Installation Parameters	7-112
MAP Subsystem	7-113
MAP Installation Procedures	7-113
MSSI Validation Requirements	7-113
Procedures for Initiating MSSI	7-114
MCS - Message Control System Version 1	7-115
Unique Parameters	7-115
Installation Parameters	7-115
MCS Procedure File	7-116
Special Notes	7-117
MSE - Mass Storage Extended Subsystem	7-118
Unique Parameters	7-118
Configuration Information	7-118

Common Deck Parameters	7-119
COMBFAS Parameters	7-119
COMXIPR Parameters	7-119
NAM5/NAM5D - Network Access Method Version 1	7-120
Configuration Information	7-120
Site-Defined User Names	7-122
Special Notes	7-123
Unique Parameters	7-124
NAM Procedure File	7-124
USER File Directives	7-125
Network Startup Master File	7-126
TITLE Statement	7-126
Parameter Records	7-126
PARAM Directive	7-126
JOB Directive	7-127
Job Skeleton Records	7-129
Network Host Product (NHP) Program Requirements	7-135
CS - Communications Supervisor	7-136
NIP - Network Interface Program	7-137
NS - Network Supervisor	7-140
NVF - Network Validation Facility	7-141
NIP5870 - 5870 Printer	7-142
Installation Procedure Messages	7-142
NOS and NOS2B - Network Operating System	7-143
Configuration Information	7-143
CALLxxx Decks	7-144
Installation Parameters	7-144
COMSACC Parameters	7-146
COMSBIO Parameters	7-147
COMSIOQ Parameters	7-147
COMSJIO Parameters	7-148
COMSLSD Parameters	7-148
COMSMLS Parameters	7-149
COMSMMF Parameters	7-149
COMSMSC Parameters	7-150
COMSMTX Parameters	7-151
COMSPFM Parameters	7-154
COMSPRO Parameters	7-157
COMSREM Parameters	7-157
COMRSRX Parameters	7-158
COMSSCD Parameters	7-159
COMSSFSS Parameters	7-159
COMSSRU Parameters	7-159
COMSSSJ Parameters	7-159
COMSVED Parameters	7-160
COMTNAP Parameters	7-160
PPCOM Parameters	7-162
DSD Parameters	7-163
819 PPU Driver Installation	7-163
NSS - NOS Scope 2 Station Facility 1	7-164

Configuration Information	7-164
Installation Procedure Messages	7-165
Spun-Off Task (SPOT) Memory Requirements	7-166
Installation Parameters	7-168
Table Size Parameters	7-168
Spooling and Recalling Time Parameters	7-169
Buffer Size Parameters	7-170
 PSU - Printer Support Utility Version 1.1	 7-171
Configuration Information	7-171
Unique Parameters	7-173
PSU Command	7-173
File Placement	7-173
NDL File Entries	7-174
 QU3 - Query Update Version 3	 7-174
 RBF5/RBF5D - Remote Batch Facility Version 1	 7-175
Special Notes	7-175
Unique Parameters	7-175
RBF Command	7-175
USER File Directives	7-176
File Placement	7-176
 RDFEX - Remote Diagnostic Facility	 7-177
 RHF - Remote Host Facility Version 1	 7-178
Configuration Information	7-178
RHF Procedure File	7-179
Hardware Configuration	7-180
Installation Parameters	7-180
NAD Microcode Initialization Parameters	7-181
NAD Microcode	7-181
 RHP - PTF/QTF File Transfer Facility	 7-182
Configuration Information	7-182
Unique Parameters	7-184
Installation Parameters	7-185
QTF Initiation Procedure Parameters	7-186
QTF Configuration Directives	7-186
 SORT5 - Sort/Merge Version 5	 7-186
Installation Procedure Messages	7-186
 TAF - Transaction Facility Version 1	 7-187
Unique Parameters	7-187
Task Library	7-188
TAF Procedure File	7-188
TAF Validation Requirements	7-188
Installation Parameters	7-189
 TCP/IP/FTP/TELNET	 7-198
Unique Parameters	7-198
Host Name and Address Definition for TCP/IP Network	7-198
NAM Startup Procedure File Changes	7-200

FTP Network Host Application Program Requirements	7-200
FTPI - File Transfer Protocol Interpreter/Server	7-201
FTPS - File Transfer Protocol Server	7-201
Network Definition Language (NDL) Requirements	7-202
TERMLIB - Terminal Library	7-204
Unique Parameters	7-204
TEXT and TEXTIO - Product TEXTS and Product TEXTS I/O	7-205
Default IPARAMS	7-205
Unique Parameters	7-206
Installation Parameters	7-207
MFT Parameter Details	7-208
Installation Procedure Messages	7-211
UPDATE (Common Memory Manager Version 1, CYBER Common Utilities, Update Version 1)	7-212

Special Product Installation Information 7

This chapter describes subsystem initiation and provides special information for customizing and configuring NOS and its products. The products are listed in alphabetical order according to the installation procedure name. If special information about a product does not exist, the product installation procedure is not listed.

NOTE

All build procedures have a number of common parameters that can be used to modify the build process. These parameters may be modified through file `COMMOD` and are discussed in detail in chapter 9.

Subsystem Initiation

Subsystem Startup Files

When you install a subsystem, make sure there is a startup procedure file for initiating the subsystem. (Table 7-1 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:

```
subffff
```

Parameter	Description
-----------	-------------

sub	The 3-character name for the subsystem (for example, <code>NAM</code>).
ffff	An optional 0- to 4-character extension to the name (for example, <code>CLSH</code>).

DECKOPL installation procedures create startup procedures for all subsystems and place them as indirect access files on the installation user name. The files are named with the appropriate 3-character subsystem name. `SYSGEN` places the released versions of all subsystem startup procedures on the system user name `SYSTEMX`. The files are private with read permission given to the installation user name. If you modify the procedures, you can use the `SYSGEN(MOVE)` procedure to move the files to the appropriate user name. Refer to chapter 8 for more information about `SYSGEN(MOVE)`.

Subsystem Enable and Disable Commands

In addition to procedure file creation and placement, you must issue `ENABLE` commands to tell the system the control point at which 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 7-1 lists the appropriate `ENABLE` commands. `DISABLE` commands are similar.

Automatic Subsystem Initiation

To automatically initiate a subsystem when you enter the `DSD AUTO` or `MAINTENANCE` commands, 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. (The released IPRDECKs contain default ENABLE commands for all ordered subsystems, except FSE and Dual State. No control points are assigned except for NAM, which is assigned control point 2.)

Table 7-1 summarizes the subsystem procedure names and the appropriate ENABLE commands necessary for initiating the subsystem.

Table 7-1. Subsystem Initiation

Installation Procedure	Subsystem	Procedure File Name	ENABLE Commands
CDCS2	CDC	CDC	ENABLE,SCP. ENABLE,CDC,n. ¹
DUAL	NVE	2	ENABLE,SCP. ENABLE,NVE,n. ¹
FSE	SMF	SMF	ENABLE,SCP. ENABLE,SMF,n. ¹
IAF	IAF	IAF, IAFTM, IAFTR	ENABLE,SCP. ENABLE,IAF.
MCS	MCS	MCS	ENABLE,SCP. ENABLE,MCS,n. ¹
MSE	MSE	MSE	ENABLE,SCP. ENABLE,MSE,n. ¹ ENABLE,MASTER MSE. ENABLE,CARTRIDGE PF STAGING.
MSSI	MAP	MAPCMD, MAPECS, MAPCH	ENABLE,SCP. ENABLE,USER EXTENDED MEMORY. ENABLE,MAP,n. ¹
NAM5/NAM5D	NAM	NAM, NAMNOGO	ENABLE,SCP. ENABLE,NAM,n. ¹
NSS	SSF	SSF	ENABLE,SCP. ENABLE,SSF,n. ¹
NOS	MAG	MAG	ENABLE,MAG.
RBF5/RBF5D	RBF	RBF is initiated by NAMI	None

1. n stands for control point number.

2. You must create your own NVE initiation procedure. Refer to the NOS/VE Software Release Bulletin and the DUAL section later in this chapter for complete information.

(Continued on next page)

Table 7-1. Subsystem Initiation

Installation Procedure	Subsystem	Procedure File Name	ENABLE Commands
<i>(Continued from previous page)</i>			
RDFEX	RDF	RDF	ENABLE,RDF.
RHF	RHF	No procedure	ENABLE,SCP. ENABLE,RHF,n. ¹
TAF	TAF	TAF	ENABLE,SCP. ENABLE,SUBCP. ENABLE,TAF,n. ¹

1. n stands for control point number.

NOTE

- RDF and MAG are always added to the deadstart file for system maintenance purposes. All other procedures are placed only on permanent files.
- MSE and RDF may require additional IPRDECK entries (refer to the NOS Version 2 Analysis Handbook).
- For more information about the IPRDECK ENABLE and DISABLE commands and subsystem startup, refer to the NOS Version 2 Analysis Handbook.
- Enable either IAF or RDF, not both. To help maintain access security to RDF, enable RDF only when needed as specified in the NOS Online Maintenance Software Reference Manual (60454200).
- 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. For more information, refer to the NOS Version 2 Analysis Handbook.

AAM2 - CYBER Record Manager Advanced Access Methods Version 2

This section describes these installation options for AAM2:

- USER File Directives
- Additional Procedures

USER File Directives

If you want to gather additional file statistics, include the Update directive *DEFINE STATS in a USER file. (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 entry point name for the first added routine must be CMPR\$11, the entry point name for the second routine 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 7-2.

Table 7-2. Format of First Three Words of Compression/Decompression Routines

Word	Bits	Contents
0	59 through 18 17 through 0	Entry point name, 6-bit display code, left-justified, zero fill. 1.
1	59 through 18 17 through 0	0. Starting address of compression code.
2	59 through 18 17 through 0	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   42/0, CMPR$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 results in 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
<pre> job command. USER, insun, password. BEGIN, PRDIN, INSTALL, PRDNAME=AAM2, DISK=0. UPDATE, K. RFL, 65000. </pre>	
<pre> COMPASS, A, I, S=TXTCRM, S=IPTEXT. </pre>	Assembles OPNMDAA and DICODAA.
<pre> SYMPL, ET=T, I, S=TXTCRM, S=IPTEXT. </pre>	Compiles OPNMDAA.
<pre> COMPASS, A. </pre>	Assembles routine being added.
<pre> RETURN, COMPILE. </pre>	Encapsulates the modified capsule OPNM\$AA (deck OPNMDAA) and the new compression capsule.
<pre> GROUP, \$AAM\$SCTL\$. CAPSULE, \$OPNM\$AA\$. CAPSULE, \$CMPR\$11\$. </pre>	
<pre> LDSET, OMIT=\$SETUP.\$/\$RM\$SYS=\$. LOAD, LGO. NOGO, NEWCAP. </pre>	User must be validated to access common files.
<pre> COMMON, SYSTEM. GTR, SYSTEM, OLD.ULIB/AAMLIB LIBEDIT, B=0. LIBGEN, F=NEW, P=AAMLIB. SYSEDIT. --eor-- *IDENT </pre>	Update directives to modify OPNMDAA.
<pre> *C OPNMDAA, DICODAA, CWEOR1, OPENDAA --eor-- </pre>	Compression/decompression routine being added (COMPASS).
<pre> --eor-- *DELETE CAP/OPNM\$AA *FILE NEWCAP *BEFORE *,CAP/* --eor-- *FILE AAMLIB --eoi-- </pre>	

APL2 - APL Version 2

The installation of APL2 consists of building APL2 and then installing APL2 permanent files. This section 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 calling the APL2 installation procedure. The file must contain only a one-line message that 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 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 installation procedure, execute these commands from your installation user name:

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

2. Run the APL2 installation procedure. When the job has finished executing, enter these commands at the system console:

```
X.DIS.
USER (SYSTEMX, SYSTEMX)
ATTACH (PFGAPL2/UN=insun)
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 automatically creates the user names and assigns default passwords. If you have a validation file, you must create user names APL0 and APL1 with the limits shown in table 7-3.

If you want to change the default passwords for the two user names (APL0 and APL1), use the MODVAL command. Refer to the NOS Version 2 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 the SYSGEN file ZZSYSGU. (Refer to SYSGEN Validations in chapter 8.)

Table 7-3. Recommended Limits for APL0 and APL1

Resource or Capability Mnemonic	User APL0 Keyboard Entry	User APL0 Converted Value	User APL1 Keyboard Entry	User APL1 Converted Value
AW	Note 1	Note 2	Note 1	Note 2
CC	77B	Unlimited	77B	Unlimited
CM	40B	2037B	40B	2037B
CN				
CP	77B	Unlimited	77B	Unlimited
CS	4B	4096	4B	4096
DB	5B	10	5B	10
DF	73B	1008	73B	1008
DS	3B	1536	2B	1024
DT				
EC	0B	0B	0B	0B
FC	7B	Unlimited	7B	Unlimited
FS	6B	192	6B	192
IS	NULL	Null	NULL	Null
LP	77B	Unlimited	77B	Unlimited
MS	6B	25088	6B	25088
MT	3	3	3	3
PA	EVEN	Even	EVEN	Even
PN				
PT	77B	Unlimited	77B	Unlimited
PW	APL0	APL0	APL1	APL1
PX	HALF	Half	HALF	Half
RO	37B	System	37B	System
RP	2	2	2	2
SC				
SL	77B	Unlimited	77B	Unlimited
SP				
TC	NORMAL	Normal	NORMAL	Normal
TL	77B	Unlimited	77B	Unlimited
TT	TTY	TTY	TTY	TTY
UP				

1. Entry is CASF, CAND, CSRP, CPWC, CLPF, CSPF, CCNR.

2. Value is 0000000000000000755.

ATF - Automated Tape Facility

This section describes the following installation options for ATF:

- Unique Parameters
- ATF Command
- File Placement

Unique Parameters

Parameter	Description
-----------	-------------

NOTRACE	To disable log file creation for ATF, specify the keyword NOTRACE on the call that executes the ATF installation procedure.
---------	---

ATF Command

The following command initiates ATF:

```
ATF.
```

The ATF command is in the ATF startup procedure file, which is part of the NAMSTRT file. Thus, ATF is started automatically with the network.

File Placement

Execution of the ATF installation procedure requires a prior build of NAM and the existence of the NAM startup master file on the installation user name. The ATF installation procedure modifies the NAM startup master file (NAMSTRT). For more information about the network startup master file, refer to the NAM5 section of this chapter.

BAM - CYBER Record Manager Basic Access Methods Version 1

This section describes the following:

- Installation Parameters
- Installation Procedure Messages

Installation Parameters

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

Parameter	Default	Significance
#CMU#	1	Specifies use of compare and move unit (CMU) instructions
#NOCMU#	1	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, the 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 the 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, 860, 870, 960, 990, 994, or 995 is not recommended.
#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.

Installation Procedure Messages

The DECKOPL installation procedure for BAM contains an update error. The following message appears in the load map:

```
0*** YANK, SELYANK, OR YANKDECK IDENT 1EP NOT KNOWN***
```

The following message appears in the dayfile:

```
1 NON-FATAL ERRORS
```

This condition is non-fatal and does not affect the generated binaries. The frequency of occurrence is relative to this product as released; any local code may change the frequency.

BASIC3 - Basic Version 3

This section describes the following:

- Installation Parameters
- Installation Procedure Messages

Installation Parameters

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	Description
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	Description
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 command to override the default); a value of 1 indicates ASCII mode (user must specify AS=0 on the BASIC command to override the default).
IP.BL	0	Flag indicating burstable listing. A value of 0 indicates nonburstable listing (user must specify BL on BASIC command to override the default); a value of 1 indicates burstable listing (user must specify BL=0 on the BASIC command to override the default).
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.

Installation Procedure Messages

The DECKOPL installation procedure for BASIC3 contains two errors. The following messages appear in the dayfile:

```
COPYL DID NOT FIND -- REL / BASSRE1
COPYL DID NOT FIND -- REL / BASSRE2
```

This condition is non-fatal and does not affect the generated binaries. The frequency of occurrence is relative to this product as released; any local code may change the frequency.

CCL - CYBER Control Language Version 1

This section describes these installation options for CCL:

- Installation Parameters
- Additional Procedures

Installation Parameters

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

Parameter	Default	Description
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. To attach the requested procedure file, the system searches the indirect access files first and then the direct access files.
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.

Parameter	Default	Description
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 two words.
IP.FP	50	Maximum number of keywords in a procedure call or procedure header directive. Maximum value is 500.
IP.FPC	10	Maximum number of characters in a keyword for a procedure call or procedure header directive. Maximum value is 10.
IP.LCS	10	Maximum number of characters in a label string. Maximum value is 10.
IP.NPV	6	Value used to calculate 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:

$$PVT = IP.NPV * IP.FP * 2$$

The PVT contains the procedure title, the parameter label/prompt, and the checklist patterns and values. The PVT also contains prompt strings from the following directives:

- CORRECT
- ENTER
- F1 through F7
- NOCLR
- NOTE
- PAGE
- PROMPT

Parameter	Default	Description
IP.PNL	50	Procedure nesting limit. Maximum value is 1023.
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.
IP.SCS	40	Maximum number of characters for default and actual values. Maximum value is 80.
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.

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 in which they occur.

NOTE

The field length allocated to CCLBRWE for prolog processing is defined in NOS deck PPCOM by symbol CCFL. Changes to CCL installation parameters or local code added to CCL may require adjustment of the value of this symbol and reassembly of NOSTEXT and VALEX.

CCP - CYBER CROSS System and Communications Control Program

This section describes the following:

- Hardware Requirements
- Configuration Information
- Released Software Variants
- Build Steps Description
- Binary Installation Option
- General Build Step Call
- CCP/CROSS Permanent Files
- Security Character Parameters
- CROSS – CROSS System Installation
- CCPPH1 – CCP Phase 1
- CCPBLB – CCP Binary Library
- CCPVAR – CCP Variant
- CCPEDIT – Edit Variant Module
- CCPLOAD – Generate CCP Load File
- CCPPURG – CCP/CROSS Installation Files Purge
- CCP System Definition
- CCP Variant Load Module Definition
- CCP Load File Definition
- CCP Extra PLs Definition
- CCP/CROSS Installation Examples

Hardware Requirements

A field length of 110000_g 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-bits 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

Configuration Information

Configuring CCP consists of generating the proper Network Load File (NLFFILE) for the 255x NPU hardware configuration and network software interfaces needed at your site. Each NPU variant within the load file is created to match the memory size of the 255x NPU and to contain the correct set of Terminal Interface Processors (TIPs) based on the communications protocols (async, HASP, X.25, etc.) used at your site. The specific variant to be used by each NPU is referenced through the VARIANT parameter on the NPU statement in the NDL file. The NLFFILE is stored as a direct access permanent file on the network administration user name. It should be a private file with read permission given to the network operations user name. NLFFILE can also be public or semiprivate.

An NLFFILE that contains several variants for different 255x NPU configurations and TIP combinations is included with the release materials. The specific entries are based on the options selected from the OIP. Refer to Released Software Variants in this section for a list of the variant names and their contents.

If you customize your own variants and load file, the load file generated by the CCPLOAD procedure must be moved from the installation user name to the network administration user name. Before the file is moved, it should be renamed from Gzzz (where zzz is value of the GN parameter from the CCPLOAD procedure call) to NLFFILE.

Released Software Variants

When you ordered CCP, you selected from one of six possible options (A through F). No matter which option you selected, you receive the CCP source program library file containing build output listings, a ready-to-use CCP load file, a sample Network Definition Language (NDL) file, and variant build output files for all CCP variants in your load file.

Table 7-4 lists the six options (A through F) available when ordering CCP and which variants are given with each. The load file received contains the variants listed for each option.

Table 7-4. Options and Variants with CCP

Option	Variants Received
A	V1F, V2F, V3F, V4F, SM1, SM2, SM3, SM4
B	V1L, V2L, V3L, V4L, V1F, V2F, V3F, V4F, SM1, SM2, SM3, SM4
C	SM1, SM2, SM3, SM4
D	SM1, SM2, SM3, SM4
E	SM1, SM2, SM3, SM4
F	SM1, SM2, SM3, SM4

Table 7-5 describes the released CCP variants.

Table 7-5. Released CCP Variants

Variant Name	NPU Size	Host Interface Program (HIP)	Link Interface Program (LIP)	Buffer Space	Terminal Interface Programs (TIPs)
V1F	96K	YES	NO	42K	ASYNCR, HASP, BISYNCR
V2F	96K	YES	NO	42K	ASYNCR, MODE4
V3F	96K	YES	NO	42K	ASYNCR, X.25 A-A, X.25 PAD
V4F	128K	YES	NO	37K	ASYNCR, HASP, MODE4, BISYNCR, X.25 A-A, X.25 PAD
V1L	96K	YES	YES	37K	ASYNCR, HASP, BISYNCR
V2L	96K	YES	YES	42K	ASYNCR, MODE4
V3L	96K	YES	YES	40K	ASYNCR, X.25 A-A, X.25 PAD
V4L	128K	YES	YES	42K	ASYNCR, HASP, MODE4, BISYNCR, X.25 A-A, X.25 PAD
SM1	65K	YES	NO	24K	ASYNCR
SM2	65K	YES	NO	19K	ASYNCR, HASP
SM3	65K	YES	NO	15K	ASYNCR, BISYNCR
SM4	65K	YES	NO	18K	ASYNCR, MODE4

When SYSGEN is executed, a compiled NDL (LCFFILE and NCFFILE) is placed on the network administration user name. The source for the NDL is placed on the network administration user name as file NDLDATA. The NDL for options A and B specifies SM1. The compiled NDL for options C through F specifies the variant name described on the OIP (option C specifies SM1, D specifies SM2, E specifies SM3, and F specifies SM4).

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 CODEPL 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.
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 Vvvvpsrout. This build step should be repeated for each variant 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) using program LFG (refer to the NOS Version 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) 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 CCP/CROSS build steps ultimately generate a NAM network load file (NLFFILE). Figure 7-1 illustrates the build step dependencies. Figure 7-2 illustrates the relationship of the load file to the release files and the other files critical to the CCP installation process. Figure 7-3 illustrates the relationship of build steps to critical files and tapes involved in CCP installation.

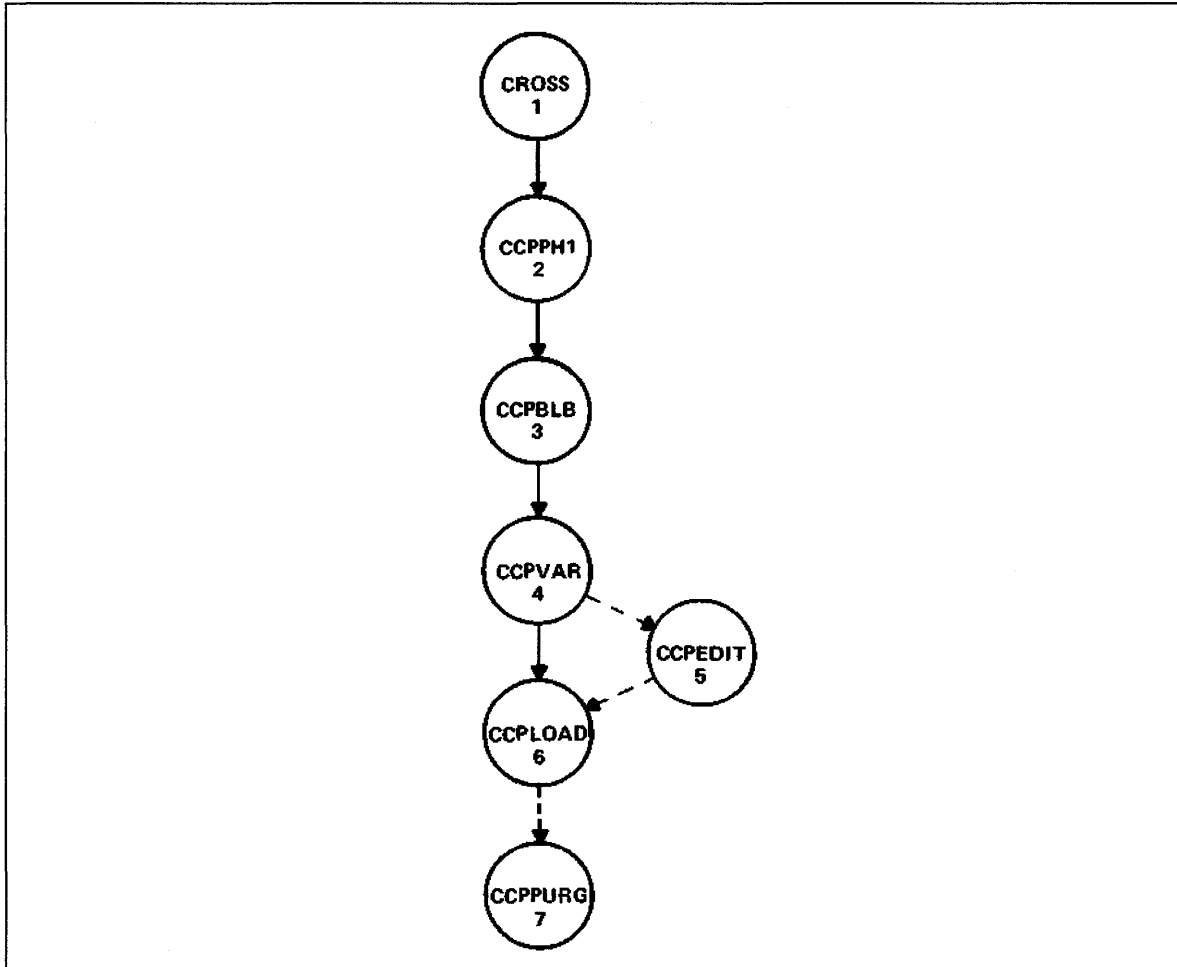


Figure 7-1. CCP/CROSS Build Step Dependencies

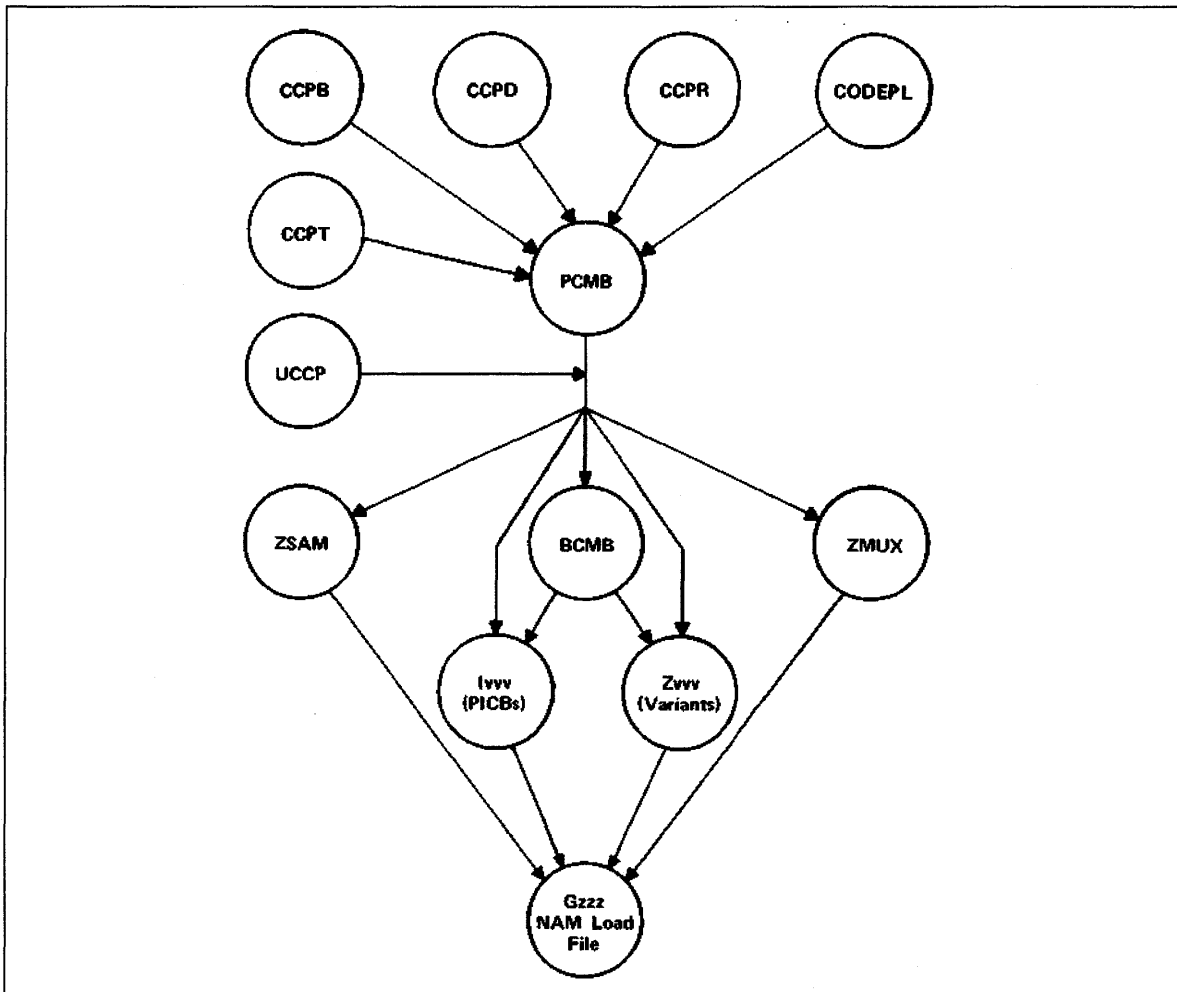
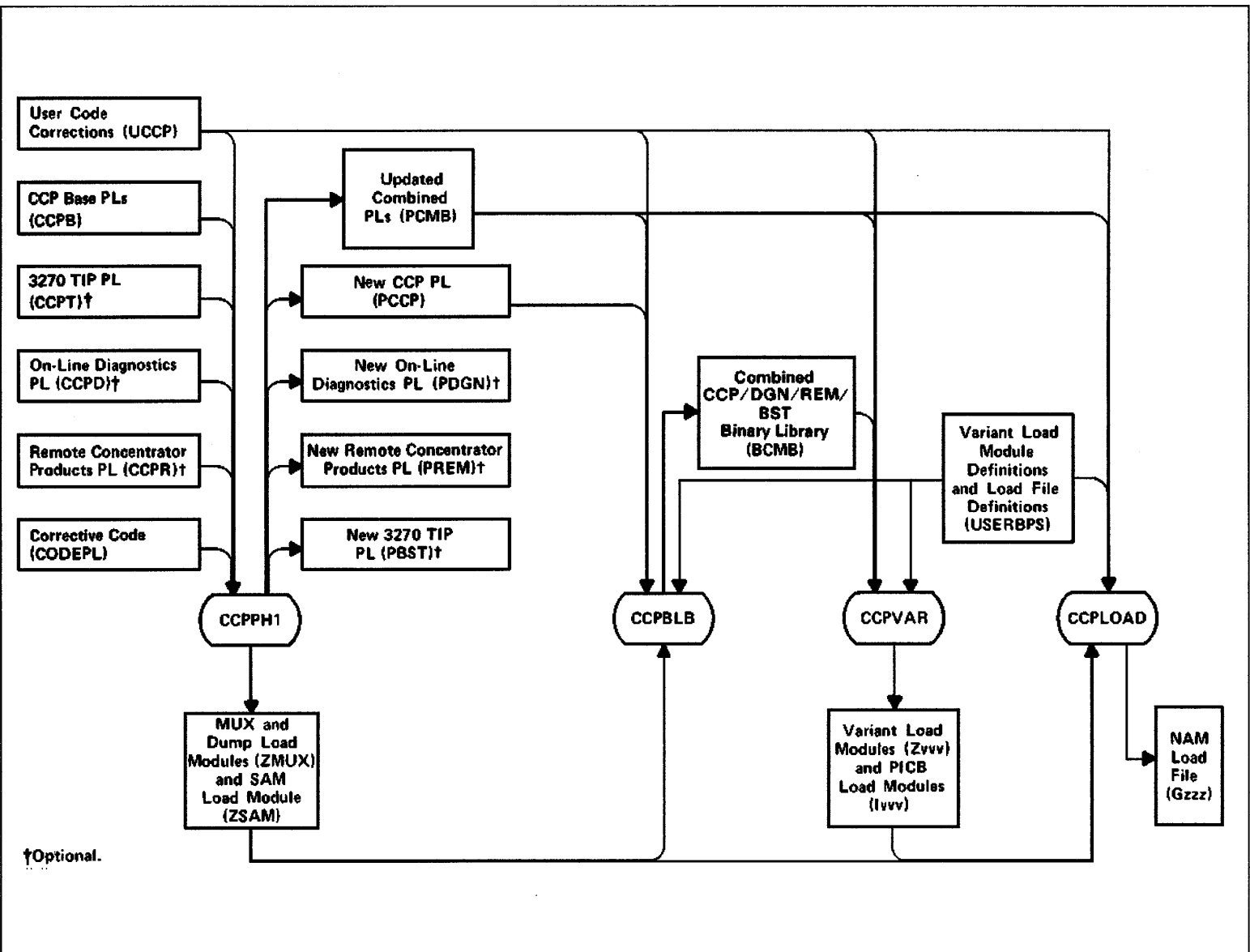


Figure 7-2. CCP File Dependencies



†Optional.

Figure 7-3. Integration of Program Libraries in CCP Installation 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 variants are described in Released Software Variants earlier in this section.)

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 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.)

The files required for a CCP binary installation were loaded by the command:

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

executed in Step 1 of chapter 6.

This installation 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. (For information about the USERBPS file, refer to User-Supplied Files under CCP/CROSS Permanent Files later in this section.)

NOTE

Complete CCP/CROSS and variant build listings (as provided by the CCPLIST=PF or VARLIST=PF options) 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 7-6 summarizes the tape and disk file requirements of the build steps.

Table 7-6. 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 USERCHG CODEPL	LCRB PCRS Addd	
2	CCPPH1	Addd	UCCP USERCHG CODEPL	ZMUX PCMB PCCP ZSAM 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		

Refer to CCP/CROSS Permanent Files later in this section for file descriptions.

The parameters used by the CCP installation procedures are listed below. The common parameters CCPLIST, NOECS, NOPURGE, and VARLIST used in the installation procedure call can be set in file COMMOD with the values described below. Refer to chapter 9 for information about modifying parameters in file COMMOD.

Parameter	Description
BSTP=bstp	Specifies whether the 3270 TIP program library is present (CCPT); used only with CCPH1. The default is BSTP=NO.
CCPLIST=option or VARLIST=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 either parameter for the CCPLOAD build step. The default is CCPLIST=NO for CROSS, CCPH1, and CCPBLB, and VARLIST=PF for CCPVAR.
	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.
	PF Listing is stored as a permanent file on disk; later it is copied to the new release file and purged from the disk.
	YES Listing is assigned to OUTPUT. For the CCPVAR build step, listing is also copied to the new release file.
DIAG=diag	Specifies whether online diagnostics are present (CCPD); used only with CCPH1. The default is DIAG=NO.
GN=zzz	Specifies load file name. The user supplies the 3-character, alphanumeric file name, which is found in the USERBPS file under the load file definitions; used only with CCPLOAD.
NEW=vvv ₂	Specifies new CCP variant name for the patched load module; used only with CCPEDIT. Supply the 3-character alphanumeric name.
NOECS	Specifies that extended memory is not used for build steps CCPH1 and CCPBLB. 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. The default is to purge these files.
OLD=vvv ₁	Specifies the CCP variant to be patched; used only with CCPEDIT. Supply the 3-character alphanumeric name.

Parameter	Description
REMT=remt	Specifies whether remote concentrator products are present (CCPR); used only with CCPPH1. The default is REMT=NO.
VN=vvv	Specifies a variant name that matches the variant name in the VRD definition in USERBPS; used only with CCPVAR.
Vx=vvv _x	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. The default is XREF=NO.
XTRAPLS=xtrapls	Specifies whether extra program libraries are to be merged into the PCMB; used only with CCPPH1. The default is XTRAPLS=NO.

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 concentrator products/3270 TIP binary library.
------	--

Corrective Code File **Description**

CODEPL	Corrective code for CCP/CROSS.
--------	--------------------------------

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 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.
LSAM	System autostart (SAM) listing.
Lvvv	Variant load module listing (vvv is variant name).

Program Libraries	Description
PBST	New 3270 TIP program library.
PCCP	New CCP program library.
PCMB	Updated combined program library.
PCRS	New CROSS program library.
PDGN	New diagnostic program library.
PREM	New remote concentrator products program library.

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

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 is 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 is 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 this listing of the permanent files.

NOTE

All CCP/CROSS user-supplied files must be permanent files under the same user name used for the build step jobs. The USERCHG file must be an indirect access permanent file. Local files of the same name are ignored.

Load Modules	Description
ZMUX	MUX load module firmware and dump bootstrap overlay.
ZSAM	SAM load module.
Zvvv	CCP variant load modules (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, under CCP/CROSS Installation Examples, later in this section.) If more than one system definition is present, only the first definition is used. The format of CCP build definitions follows:

```
keyword1=value1, ..., keywordn=valuen.
```

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

```
keyword1=value1/.../valuen.
```

This is equivalent to the following:

```
keyword1=value1, ..., keyword1=valuen.
```

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.

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 (LCFFILE and NCFFILE) 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 Version 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. The secure login is not activated for any sub-TIP for which a security character is not specified (that is, value equals \$00).

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

Parameter	Default Security Character	Terminal
SCA2741	\$00	Asynchronous 2741 terminals
SCAN2741	\$00	Asynchronous non-2741 terminals
SCB2780	\$00	IBM 2780 terminals
SCB3270	\$00	IBM 3270 terminals
SCB3780	\$00	IBM 3780 terminals
SCHPOST	\$00	HASP postprint terminals
SCHPRE	\$00	HASP preprint terminals
SCMD4A	\$00	Mode 4A terminals
SCMD4C	\$00	Mode 4C 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 the programs needed for subsequent CCP build steps. Refer to General Build Step Call, earlier in this section, for descriptions of the parameters.

```
BEGIN, CROSS, INSTALL, CCPLIST=option, NOPURGE.
```

The CROSS build step uses the following files for input.

File	Description
CODEPL	CROSS corrective code, if any, that affects the resulting CROSS binaries but is not placed in the program library on the output file.
CRSSpsrin	CROSS release file.
UCRS	Optional site corrective code (refer to CCP/CROSS Permanent Files, earlier 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.

The CROSS build step creates the following output files.

File	Description
AASM	CROSS macro assembler.
ACYP	CYBER 180, CYBER 170, CYBER 70, and 6000 Computer Systems Pascal compiler.
AEDT	MPEDIT 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.
CRSSpsrout	New CRSS program library.
LCRB	CROSS system listings (if requested).

Installation Parameters for CROSS

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 77000₈ 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 products 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=bstp, XTRAPLS=xtrapls,
CCPLIST=option, NOECS, NOPURGE.
```

The CCPPH1 build step uses the following input files.

File	Description
CCPBpsrin	CCP release program library.
CCPDpsrin	Diagnostic release program library (if DIAG=YES).
CCPRpsrin	Remote concentrator products release program library (if REMT=YES).
CCPTpsrin	3270 TIP release program library (if BSTP=YES).
CODEPL	CCP corrective code, if any.
UCCP	Optional user corrective code. Refer to User-Supplied Files in CCP/CROSS Permanent Files, earlier in this section.

CCPPH1 generates the following output files.

File	Description
AALK	Autolink program.
AEXP	Build parameters expand program.
CCPDpsrout	New diagnostic program library (if DIAG=YES).
CCPRpsrout	New remote concentrator products program library (if REMT=YES).
CCPTpsrout	New 3270 TIP program library (if BSTP=YES).
LIMC	Expand and Autolink program listings (if requested).
LMFB	CCP list file (if requested).
	File 1 Multiplexer firmware.
	File 2 Dump bootstrap overlay.
LSAM	System autostart (SAM) listing (if requested).
PBST	New 3270 TIP program library (if BSTP=YES).
PCCP	New CCP program library.
PCMB	Updated combined program library.
PDGN	New diagnostic program library (if DIAG=yes).
PREM	New remote concentrator products program library (if REMT=yes).
SMUX	Symbol table for dump bootstrap.
ZMUX	CCP load module.
	File 1 Multiplexer firmware.
	File 2 Dump bootstrap overlay.
ZSAM	SAM load module.

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, CCPLIST=option, XREF=xref, NOECS, NOPURGE.
```

CCPBLB requires the following input files.

File	Description
AALK	Autolink program.
AASM	CROSS macro assembler.
ACYP	CYBER Computer Systems Pascal compiler (if XREF=YES).
AEXP	Build parameters expand program.
AFMT	Pascal binary output formatter program.
AMAC	Macro assembler text file.
APAS	MP17 Pascal compiler.
AXRF	Pascal cross-reference program (if XREF=YES).
PCMB	Updated combined program library.
SMUX	Symbol table for dump bootstrap.
UCCP	Optional user corrective code. Refer to User-Supplied Files under CCP/CROSS Permanent Files, earlier in this section.
USERBPS	User variant build parameters file. Refer to User-Supplied Files under CCP/CROSS Permanent Files, earlier in this section.

CCPBLB produces these output files.

File	Description
BCMB	Combined CCP/diagnostics/remote concentrator products/3270 TIP binary library.
CCPBpsrout	New CCP program library.
LFCA	Full compile assembly listings (if requested).
	File 1 Assembly source listing.
	File 2 Pascal source and object listing.

CCPVAR - CCP Variant

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

```
BEGIN, CCPVAR, INSTALL, VARLIST=option, VN=vvv, NOPURGE.
```

CCPVAR requires the following input files.

File	Description
AALK	Autolink program.
AEDT	MPEDIT program.
AEXP	Build parameters expand program.
AFMT	Pascal binary output formatter program.
ALNK	MPLINK program.
APAS	MP17 Pascal compiler.
BCMB	Combined CCP/diagnostics/remote concentrator products/3270 TIP binary library.
PCMB	Updated combined program library.
SMUX	Symbol table for dump bootstrap.
UCCP	Optional user corrective code. Refer to User-Supplied Files under CCP/CROSS Permanent Files, earlier in this section.
USERBPS	User variant build parameters file. Refer to User-Supplied Files under CCP/CROSS Permanent Files, earlier in this section.

CCPVAR generates the following output files (vvv is the variant name).

File	Description
Ivvv	CCP program initiation control block load modules.
Lvvv	Variant load module listing (if requested).
Svvv	Symbol table for load module Zvvv.
Vvvvpsrout	Variant release file.
Zvvv	CCP variant load module.

CCPEDIT - Edit Variant Module

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 the 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=vvv1, NEW=vvv2.
```

CCPEDIT requires four input files.

File	Description
AEDT	MPEDIT program.
Svvv ₁	Symbol table for load module Zvvv ₁ .
UEDZ	Optional direct or indirect access permanent file of MPEDIT directives to patch a CCP variant load module. Refer to User-Supplied Files under CCP/CROSS Permanent Files, earlier in this section.
Zvvv ₁	CCP variant load module vvv ₁ .

CCPEDIT produces two output files.

File	Description
Svvv ₂	Copy of symbol table Svvv ₁ .
Zvvv ₂	New CCP variant load module reflecting patch code.

CCPLOAD - Generate CCP Load File

Based on user-supplied load file definitions in 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
AEXP	Build parameters expand program.
Ivvv	CCP PICB load modules.
PCMB	Updated combined program library. Refer to User-Supplied Files under CCP/CROSS Permanent Files, earlier in this section.
UCCP	Optional user corrective code. Refer to User-Supplied Files under CCP/CROSS Permanent Files, earlier in this section.
USERBPS	CCP load file definitions file supplied by user. Refer to User-Supplied Files under CCP/CROSS Permanent Files, earlier in this section.
ZMUX	MUX firmware and dump bootstrap overlay.
ZSAM	SAM load module.
Zvvv	CCP variant load modules.

CCPLOAD generates one output file.

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

If the released version of the NS job skeleton JOBNS is used (refer to the NAM5 section in this chapter), rename Gzzz file as NLFFILE and move NLFFILE to the network administration user name. NLFFILE should be a direct access file with read permission given to the network operations user name.

CCPPURG - CCP/CROSS Installation Files Purge

The following optional build step purges all the permanent files created by the CCP/CROSS installation process. This step does not purge the user-supplied files (USERCHG, USERBPS, UCCP, UCRS, and UEDZ associated with the variants in the build call), source PLs, or the CCP load file created by CCPLOAD (Gzzz). Refer to General Build Step Call, earlier in this section, for descriptions of the parameters.

```
BEGIN, CCPPURG, INSTALL, V1=vvv1, . . . , Vn=vvvn.
```

This step requires no input files and produces no output files.

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=v ₁ /v ₂ /v ₃	Specifies options if present.
v_i	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 in the account dayfile and the error log file, are compiled.
R	Remote concentrator products are present.
T	Support modules for the test utility program (TUP) and CONSOLE are compiled. (TUP is an unsupported product.)

Keyword	Description
TS=t ₁ /.../t _n	Specifies which Terminal Interface Programs (TIPs) 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.
M	Mode 4 TIP is included.
T	3270 TIP is included.
XA	X.25 TIP and application-to-application sub-TIP are included.
XP	X.25 TIP and PAD sub-TIP are included. Specify this TIP for any variant that executes in an NPU connected to a packet switching network.
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 prior to the last ends with comma. The last line must end with a period.

The format follows:

VRD=vvv,VT=v1/v2/v3,SZ=xK,TS=t1/.../tn.

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 Link Interface Program (LIP).
L	Local; includes HIP and LIP.
R	Remote; includes LIP but no HIP.

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

Keyword	Description																						
SZ=xK	Specifies variant memory size: 65K, 81K, 96K, or 128K (x is a 2- or 3-digit number).																						
TS=t ₁ /.../t _n	Specifies which Terminal Interface Programs (TIPs) are to be included in this variant. TS can assume up to 10 different order-independent values.																						
	<table border="1"> <thead> <tr> <th>t_i</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Asynchronous TIP is included. This TIP supports ASCII terminals, APL character sets and IBM 2741 terminals.</td> </tr> <tr> <td>B</td> <td>Binary synchronous communications (BSC) TIP is included. This TIP supports the IBM 2780 and IBM 3780 terminals.</td> </tr> <tr> <td>H</td> <td>HASP TIP is included.</td> </tr> <tr> <td>M</td> <td>Mode 4 TIP is included.</td> </tr> <tr> <td>T</td> <td>3270 TIP is included.</td> </tr> <tr> <td>XA</td> <td>X.25 TIP and application-to-application sub-TIP are included.</td> </tr> <tr> <td>XP</td> <td>X.25 TIP and PAD sub-TIP are included. Specify this TIP for any variant that executes in an NPU connected to a packet switching network.</td> </tr> <tr> <td>1</td> <td>User TIP1 is included.</td> </tr> <tr> <td>2</td> <td>User TIP2 is included.</td> </tr> <tr> <td>3</td> <td>User TIP3 is included.</td> </tr> </tbody> </table>	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.	M	Mode 4 TIP is included.	T	3270 TIP is included.	XA	X.25 TIP and application-to-application sub-TIP are included.	XP	X.25 TIP and PAD sub-TIP are included. Specify this TIP for any variant that executes in an NPU connected to a packet switching network.	1	User TIP1 is included.	2	User TIP2 is included.	3	User TIP3 is included.
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.																						
M	Mode 4 TIP is included.																						
T	3270 TIP is included.																						
XA	X.25 TIP and application-to-application sub-TIP are included.																						
XP	X.25 TIP and PAD sub-TIP are included. Specify this TIP for any variant that executes in an NPU connected to a packet switching network.																						
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=vvv1/.../vvvn.
```

Keyword	Description
LFD=zzz	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=vvv _i	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=ppp1/.../pppn.
```

Keyword	Description
PLS=ppp _i	Identifies entry as an extra PL definition and specifies which user-supplied PLs should be merged into PCMB. The parameter ppp _i is the name of the file to be merged with CCP. This entry is used by the CCPPH1 step.

CCP/CROSS Installation Examples

Examples follow which illustrate installation of CCP in two network configurations: three NPUs (two local NPUs and one remote NPU) and a multihost configuration with three hosts, two NPUs, and an X.25 line.

- All examples may use the following input files.

Files	Description
UCCP	Required for user-suggested or PSR code for CCP.
UCRS	Required for user-suggested or PSR code for CROSS.
USERBPS	Required for CCPBLB, CCPVAR and CCPLOAD; contains CCP system definitions, variant definitions and load file definitions.

- 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: Three NPUs

Figure 7-4 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 assignments and/or NDL name for major components in the network as chosen for this example. In the configuration shown in figure 7-4, the following conditions apply:

- 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 7-4.

1. Ensure that the required files and tapes are available. Refer to figure 7-5 for appropriate USERBPS definitions, EQPDECK entries, and NDL source input.
2. Install CROSS.

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

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

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

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

This step requires CCPBpsrin, CCPDpsrin, and CCPRpsrin. CCPLIST=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,CCPLIST=BOTH,XREF=YES.
```

This step requires CCPBpsrin. CCPLIST=BOTH routes the listings to the printer and stores them as permanent files during the installation procedure. At the end of the procedure, 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.

```
BEGIN,CCPVAR,INSTALL,VARLIST=PF,VN=VNA.
```

VARLIST=PF stores the listings as permanent files during the installation procedure. At the end of the procedure, 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.

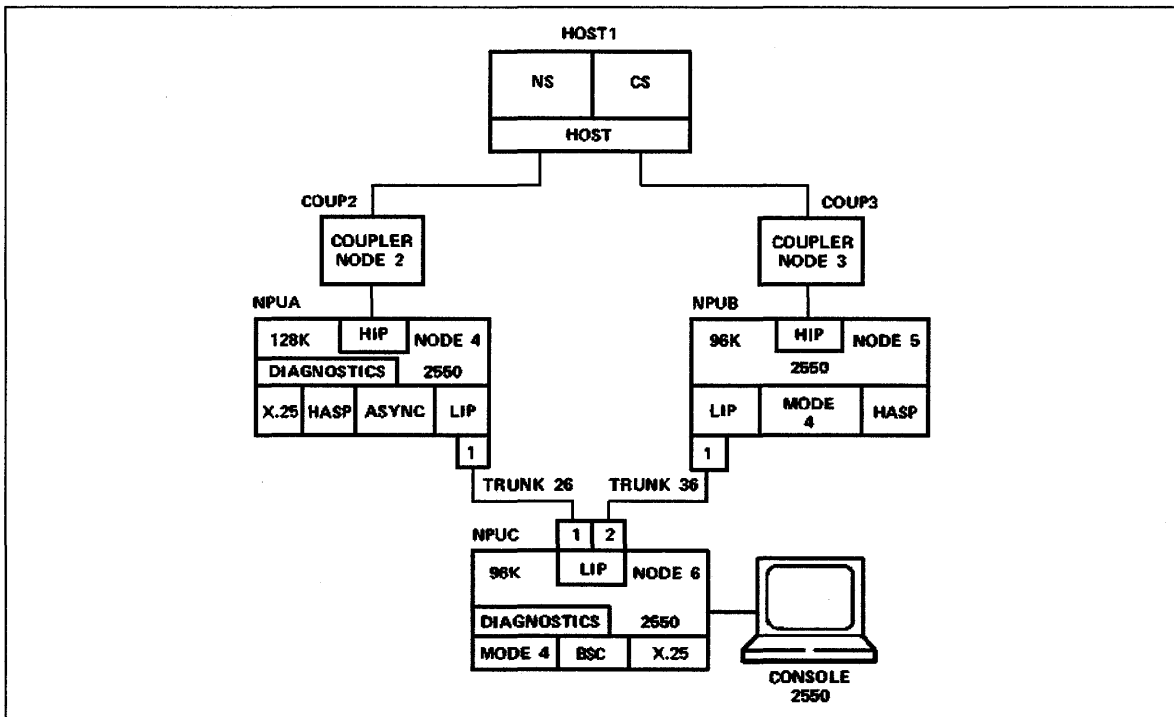


Figure 7-4. Network Configuration - Example 1

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.

```

    BEGIN, CCPVAR, INSTALL, VARLIST=PF, VN=RMC.
    
```

The load module file name is ZRMC and the PICB file name is IRMC. VARLIST=PF stores the listings as permanent files during the installation procedure. At the end of the procedure, the files are copied to the new release file 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.

9. Execute this build step only if you want to purge all permanent files created by the CCP/CROSS installation process.

```

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

```

*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       e d           g i k
VRD=VNA,VT=L/D,SZ=128K,TS=A/H/XP.
*   b       e           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/E/XP.
*
*LOAD FILE DEFINITION IS
*
*   n       a   b   c
LFD=EX3,LM=VNA/VNB/RMC.

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.
                                NDL Source Input

SUPLINK LLNAME=LL35.

COUP3: COUPLER NODE=3, HNAME=HOST1.

LL35: LOGLINK NCNAME=NPUB.

LL36: LOGLINK NCNAME=NPUC.

```

Figure 7-5. USERBPS Definitions, NDL Source Input, and EQPDECK Entries for Example 1 (Continued)

```

                                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=0FF.      EQPDECK Entries
EQ42=NP,ST=ON,EQ=7,PI=1,CH=5,ND=3,SA=0FF.

```

Figure 7-5. USERBPS Definitions, NDL Source Input, and EQPDECK Entries for Example 1

Example 2: Three Hosts, Two NPUs, X.25 Line

Figure 7-6 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 7-6, the following conditions apply:

- NPUA has four TIPs, a HIP, and an X.25 line. The latter two programs are required for the coupler and trunk respectively.
- NPUB has three TIPs, a HIP, and an X.25 line. The latter two programs are required for the coupler and trunk respectively.
- HOST1 has NS only.
- HOST2 has CS only.
- HOST3 has NS and CS.
- PTF and QTF are installed on all three hosts.
- Host-to-host and host-to-NPU (X.25) are defined for PTF/QTF transfers.

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 7-6.

1. Ensure that the required files and tapes are available. Refer to figure 7-7 for the appropriate USERBPS definitions, NDL source input, and EQPDECK entries.
2. Follow steps 2 through 6 in example 1.
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 all permanent files created by the CCP/CROSS installation process.

```

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

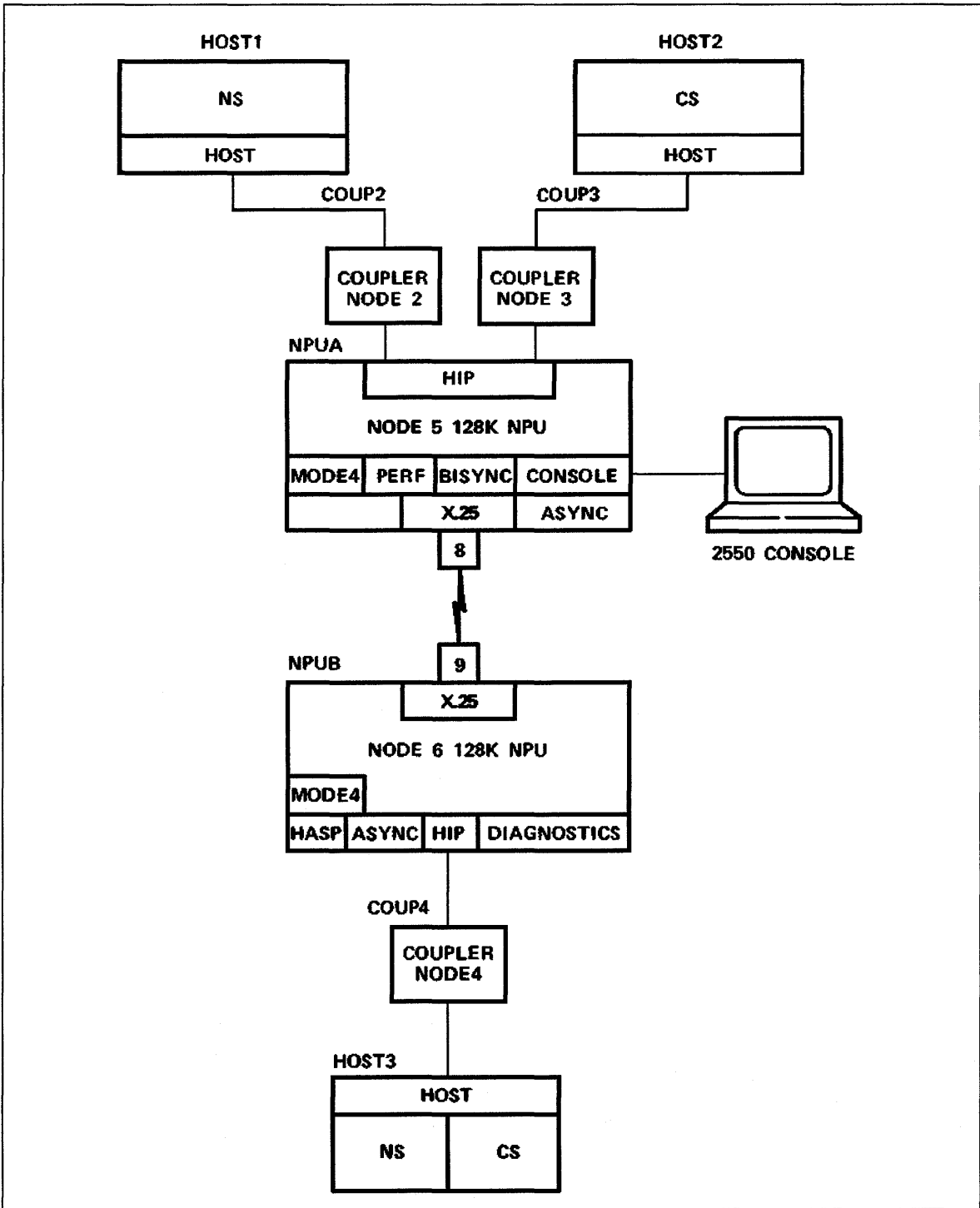


Figure 7-6. Network Configuration - Example 2

```

*   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
*       XP     X25 PAD
*
*   c d e     g h i j k m
SYS=C/D/P,TS=A/B/H/M/XA/XP.*
*   VARIANT DEFINITIONS
*
*       NPU A
*
*       a     c     e             g h j k m
*VRD=VNA,VT=C/F/P,SZ=128K,TS=A/B/M/XA/XP.
*
*       NPU B
*
*       b     d             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.
*
*

```

USERBPS Definitions

Figure 7-7. USERBPS Definitions, NDL Source Input and EQPDECK Entries for Example 2 (Continued)


```

NCF2P2:NFILE.

                                a
NPUA:NPU NODE=5, VARIANT=VNA.
SUPLINK LLNAME=LL35.
SUPLINK LLNAME=LL25.
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=XAA.

                                b
NPUB:NPU NODE=6, VARIANT=VNB.                                NDL Source Input
SUPLINK LLNAME=LL46.
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=XAA.

END.

```

Figure 7-7. USERBPS Definitions, NDL Source Input and EQPDECK Entries for Example 2 (Continued)

```

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,PRU,NETXFR,PRIV.

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

INCALL,FAM=0,UNAME=netopun,ANAME=QTFS,DBL=7,ABL=7,DBZ=1024.
OUTCALL,NAME1=QTFS,PID=MO2,SNODE=2,DNODE=3,DBL=7,ABL=7,DBZ=1024.
INCALL,FAM=0,UNAME=netopun,ANAME=PTFS,DBL=7,ABL=7,DBZ=1024.
OUTCALL,NAME1=PTFS,PID=MO2,SNODE=2,DNODE=3,DBL=7,ABL=7,DBZ=1024.
INCALL,FAM=0,UNAME=netopun,ANAME=QTFS,PORT=8,SNODE=5,DNODE=2,DBZ=1024,DBL=7,
ABL=7,UBZ=1024,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=QTFS,PID=MO3,SNODE=2,DNODE=5,SHOST=2D3033,PORT=8,DHOST=4,
DBZ=1024,DBL=7,ABL=7,UBZ=1024,UBL=7,WS=7,DPLS=1024.
INCALL,FAM=0,UNAME=netopun,ANAME=PTFS,PORT=8,SNODE=5,DNODE=2,DBZ=1024,DBL=7,
ABL=7,UBZ=1024,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=PTFS,PID=MO3,SNODE=2,DNODE=5,SHOST=2D3033,PORT=8,DHOST=4,
DBZ=1024,DBL=7,ABL=7,UBZ=1024,UBL=7,WS=7,DPLS=1024.

```

Figure 7-7. USERBPS Definitions, NDL Source Input and EQPDECK Entries for Example 2 (Continued)

```

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=netopun,ANAME=QTFS,DBL=7,ABL=7,DBZ=1024.
OUTCALL,NAME1=QTFS,PID=MO1,SNODE=3,DNODE=2,DBL=7,ABL=7,DBZ=1024.
INCALL,FAM=0,UNAME=netopun,ANAME=PTFS,DBL=7,ABL=7,DBZ=1024.
OUTCALL,NAME1=PTFS,PID=MO1,SNODE=3,DNODE=2,DBL=7,ABL=7,DBZ=1024.
INCALL,FAM=0,UNAME=netopun,ANAME=QTFS,PORT=8,SNODE=5,DNODE=3,DBZ=1024,DBL=7,
    ABL=7,UBZ=1024,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=QTFS,PID=MO3,SNODE=3,DNODE=5,SHOST=2D3033,PORT=8,DHOST=4,
    DBZ=1024,DBL=7,ABL=7,UBZ=1024,UBL=7,WS=7,DPLS=1024.
INCALL,FAM=0,UNAME=netopun,ANAME=PTFS,PORT=8,SNODE=5,DNODE=3,DBZ=1024,DBL=7,
    ABL=7,UBZ=1024,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=PTFS,PID=MO3,SNODE=3,DNODE=5,SHOST=2D3033,PORT=8,DHOST=4,
    DBZ=1024,DBL=7,ABL=7,UBZ=1024,UBL=7,WS=7,DPLS=1024.

```

Figure 7-7. USERBPS Definitions, NDL Source Input and EQPDECK Entries for Example 2 (Continued)

```

LCFFFILE:LFILE

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

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

```

Figure 7-7. USERBPS Definitions, NDL Source Input and EQPDECK Entries for Example 2

CDCNET - Control Data Distributed Communications Network

This section describes how to install and configure CDCNET. CDCNET is composed of two subproducts: DCNS and CHA1. DCNS is a binary-supported product that consists of the ANACD, DI, MANCC, and NPA software. It is updated through a CDCNET Batch Critical Update (BCU), which is described in chapter 5. CHA1 consists of the NAM applications and utilities that interface with the DCNS software.

NOTE

CDCNET and its separately priced TIPs are supported in 64-character set only.

General CDCNET information is documented in the following subsections:

- Configuration Information
- Installing CDCNET for the First Time
- Upgrading CDCNET
- Activating a New Level of CDCNET
- Installing a CDCNET Separate TIP
- Installing CDCNET for MDI Reset Support Only
- Installation Wrapup Activities
- NPA Database Maintenance

Information relating to the CHA1 portion of the CDCNET software is documented in the following subsections:

- CDCNET Host Application (CHA1) Product Content
- Unique Parameters
- USER File Directives
- Installation Parameters
- Message Templates
- NAM Startup Procedure File Changes
- CDCNET Network Host Application Program Requirements
- Network Definition Language (NDL) Requirements

Configuration Information

- CDCNET Configuration Files and Procedures. CDCNET uses several different types of configuration files and procedures:
 - Device Interface (DI) system configuration procedures define the hardware and software attributes of the different types of DIs.
 - Terminal User Procedures (TUPs), Terminal Definition Procedures (TDPs), and Load Procedures (LPs) are used to specify additional information about lines and terminal and batch devices in addition to the information specified in the system configuration procedures.
 - Exception List file controls the loading of non-host-resident DIs. Loading of host-resident DIs is controlled by the INITDCN NOS permanent file.

The DI system configuration procedures, TDPs, TUPs and LPs can be maintained in individual files or libraries. Refer to the CDCNET Configuration Guide for more information.

During permanent file installation, SYSGEN automatically creates configuration files that allow you to load and configure a DI and log in to a terminal for completing installation.

All CDCNET configuration files reside on the network administration user name except for the network directory file (NETDIR) which resides on the network operations user name. You can log in to the network administration user name to perform the configuration of CDCNET. The user name has been given write permission to the NETDIR file on the network operations user name. This permission allows you to update this file from an interactive terminal.

For more information on configuring CDCNET, refer to the CDCNET Configuration Guide.

- Local Configuration File. The LCF portion of the NDL defines the applications (with APPL statements) that allow CDCNET to interface to NAM. INCALL statements define the connections between the DI software and the NAM applications. The required statements are in the released sample NDL file named NDLDATA. The statements are documented in Network Definition Language Requirements later in this section. For information on compiling a new NDL, refer to the NAM5 section of this chapter.
- EQ EQPDECK entry for CDCNET. This entry defines the connection of each Mainframe Device Interface (MDI) or Mainframe Terminal Interface (MTI) to the mainframe. Optionally, the ND and NT parameters are referenced in CDCNET DI configuration files. For more information about the EQ EQPDECK entry for CDCNET, refer to the Deadstart Deck section of the NOS Version 2 Analysis Handbook.

Installing CDCNET for the First Time

Perform the following instructions if you are installing CDCNET for the first time as part of an upgrade or component order installation.¹ One step is performed from the system console and the rest of the steps are performed from an interactive terminal.

Before beginning the steps below, make sure that the network administration user name has been created. Also, have the permanent file tapes available for mounting.

The values `insun`, `netopun`, and `netadun` should be substituted throughout the following steps with the user names that the site is using for the installation user name, network operations user name, and the network administration user name, respectively. If you wish to use the Control Data default values, replace `insun` with `INSTALL`, `netopun` with `NETOPS`, and `netadun` with `NETADMN`.

1. Install CDCNET files to the installation user name.
 - a. Log in to IAF under the installation user name. If you are installing CDCNET as part of a component order, skip step b.
 - b. Make the newly released SYSGEN procedures local to your job by entering:

```
GET, SYSGEN.
```

- c. Load CDCNET permanent files from the permanent file tape by entering:

```
SYSGEN, RECLAIM, PFGDCNS, PFGCHA1.
```

The permanent file tapes are requested.

- d. If you are not changing the default installation user name, network operations user name, or the network administration user name, skip to step e. To change these user names, enter the following command:

```
BEGIN, CHASUN, PFGDCNS, insun, netopun, netadun.
```

Replace `insun` with the installation user name, `netopun` with the network operations user name, and `netadun` with the network administration user name. By default, these user names are `INSTALL`, `NETOPS`, and `NETADMN`, respectively.

If you changed the network administration user name from `NETADMN`, a local file named `LGO` will be created. This contains updated versions of procedures `MANCC`, `ANACD`, and `NPA`; that is, these procedures will access files from the new network administration user name. File `LGO` must be `LIBEDITed` into files `GLOBLIB` and `PRODUCT`. `LIBEDITing` into `GLOBLIB` ensures that an analyst using these procedures gets files from the new user name. `LIBEDITing` into `PRODUCT` ensures that the new deadstart tape will contain the updated versions.

- e. Create the CDCNET installation procedure file by entering:

```
BEGIN, INSPL, PFGDCNS.
```

¹ If you are installing CDCNET in a multihost network, refer to Installing CDCNET for MDI Reset Support Only, later in this section.

- f. Load the CHA1 message template file, HTF_id_psout (where id = A, psout = NOS release level) to disk on the installation user name by entering:

```
BEGIN, INITIAL, DCNPLIB, insun.
```

A message is displayed at your terminal indicating the version level of CDCNET being installed.

- g. Log out of the installation user name.
2. Create the network directory file (NETDIR) on the network operations user name by performing these steps from the system console.

- a. Enter these commands:

```
X.DIS.  
USER, netopun, password.  
ATTACH, DCNPLIB/UN=insun.  
BEGIN, INITIAL, DCNPLIB, netopun.
```

- b. The network administration user name is given access to the directory. Once this procedure completes, enter:

```
DROP.
```

3. Load CDCNET files to the network administration user name by performing the following steps from an interactive terminal:

- a. Log in to IAF under the network administration user name.

- b. Enter the following commands:

```
ATTACH, DCNPLIB/UN=insun.  
BEGIN, INITIAL, DCNPLIB, netadun.
```

This step requests the permanent file tapes, loads all DCNS software to disk (including configuration files), and initializes the NPA databases. A message is displayed to indicate the version level of CDCNET being installed. This step also displays NETFM output as the CDCNET files are added to the directory. Check the NETFM output on the screen to ensure that all CREATE directives completed normally. When the procedure completes, all new CDCNET software files have been loaded to the network administration user name.

4. Define the system ID of the MDI or MTI connected to your mainframe using the procedure ADDDI (ADD_DEVICE_INTERFACE). If you are not executing this step from the same terminal session as step 3, log in to the network administration user name and execute the commands below to access the installation tools required in file GLOBLIB on the installation user name:

```
ATTACH, GLOBLIB/UN=insun.  
LIBRARY, GLOBLIB/A.
```


You can now use NETPLM, NETFM, MANCC, and so forth, for configuring CDCNET.

- a. Define the system IDs for each DI in the network using procedure ADDDI (ADD_DEVICE_INTERFACE). Enter the following commands from an interactive terminal logged in to the network administration user name:

```
BEGIN, ADDDI, DCNPLIB, type, si.
```

Parameter	Description
type	Type of DI (MDI or MTI).
si	Last six digits of the DI's system identifier. The DI system identifier is a 12-digit number printed on a tag attached to the inside of the DI cabinet. Note that the tag contains a 4-digit checksum appended to the 12-digit system identifier. Do not include the checksum in this parameter.

Check the NETPLM output on the screen to be sure the configuration file is successfully added to the network directory file.

If you configured an MTI in step a, skip step b.

- b. Repeat the ADDDI procedure call for each TDI you define. Specify the DI's system identifier (si):

```
BEGIN, ADDDI, DCNPLIB, TDI, si.
```

After completing the procedures described in this section, refer to the CDCNET Configuration Guide to modify configuration files for your site. GLOBLIB must be available in order to use NETPLM, NETFM, MANCC, and so forth.

Upgrading CDCNET

Perform the following instructions if you are upgrading CDCNET as part of an upgrade installation.² If you are upgrading CDCNET to a new BCU level, follow the instructions in chapter 5.

The values insun, netopun, and netadun should be substituted throughout the following steps with the user names that the site is using for the installation user name, network operations user name, and the network administration user name, respectively. If you wish to use the Control Data default values, replace insun with INSTALL, netopun with NETOPS, and netadun with NETADMN.

² If you are installing CDCNET in a multihost network, refer to Installing CDCNET for MDI Reset Support Only, later in this section.

NOTE

If you are changing the default Control Data installation user names (in particular, NETOPS and NETADMN), note the following:

- If you change the network operations user name:
 1. Move the network directory file NETDIR to the new user name and update its permissions, if any.
 2. Update the file permissions of the files on user name NETADMN (for example, ELIST) to give READ permission to the new user name. (If you are also changing the network administration user name, update the file permissions after moving the files from NETADMN.)
- If you change the network administration user name:
 1. Move all files currently on NETADMN to the new user name.
 2. Update the directory entries for these NETADMN files to specify the new user name. For example, to update the directory for file ELIST, execute the following commands from the network administration user name:

```
NETFM, UN=netopun, Z. /DELETE, PF=ELIST, NF=EXCEPTION_LIST, UN=NETADMN
NETFM, UN=netopun, Z. /CREATE, PF=ELIST, NF=EXCEPTION_LIST
```

A similar pair of commands should be executed for each file that is listed in the network directory.

1. Install CDCNET files to the installation user name.
 - a. Log in to IAF under the installation user name.
 - b. Make the newly released SYSGEN procedures local to your job by entering:


```
GET, SYSGEN.
```
 - c. Load CDCNET permanent files from the permanent file tape by entering:


```
SYSGEN, RECLAIM, PFGDCNS, PFGCHA1.
```

The permanent file tapes are requested.
 - d. If you are not changing the default installation user name, network operations user name, or the network administration user name, skip to step e. To change these user names, enter the following command:

```
BEGIN, CHASUN, PFGDCNS, insun, netopun, netadun.
```

Replace insun with the installation user name, netopun with the network operations user name, and netadun with the network administration user name. By default, these user names are INSTALL, NETOPS, and NETADMN, respectively.

If you changed the network administration user name from NETADMN, a local file named LGO will be created. This contains updated versions of procedures MANCC, ANACD, and NPA; that is, these procedures will access files from the new network administration user name. File LGO must be LIBEDITed into files GLOBLIB and PRODUCT. LIBEDITing into GLOBLIB ensures that an analyst using these procedures gets files from the new user name. LIBEDITing into PRODUCT ensures that the new deadstart tape will contain the updated versions.

- e. Create the CDCNET installation procedure file by entering:

```
BEGIN, INSPL, PFGDCNS .
```

- f. Load the CHA1 message template file, HTF_id_psrou (where id=A, psrou=NOS release level) to disk on the installation user name by entering:

```
BEGIN, UPGRADE, DCNPLIB, insun.
```

A message is displayed at your terminal indicating the version level of CDCNET being installed.

- g. Log out of the installation user name.

2. Load CDCNET files to the network administration user name. Perform the following steps from an interactive terminal:

- a. Log in to IAF under the network administration user name.

- b. Enter the following commands:

```
PURGE, RECLDB/NA.  
ATTACH, DCNPLIB/UN=insun.  
BEGIN, UPGRADE, DCNPLIB, netadun.
```

This step requests the permanent file tapes and loads DCNS software to disk. A message is displayed to indicate the version level of CDCNET being installed. This step also displays NETFM output as the CDCNET files are added to the directory. Check the NETFM output on the screen to ensure that all CREATE directives complete normally. When the procedure completes, all new CDCNET software files have been loaded to the network administration user name. Note that no site-modifiable configuration files are loaded.

You have completed installing the new level of CDCNET software. The instructions in chapter 3 will direct you back to this section when it is time to activate a new CDCNET and perform wrapup activities.

Activating a New Level of CDCNET

If you installed CDCNET for MDI Reset Support Only, there is no need to activate the new level of CDCNET because there is no CDCNET software to load.

When a new version of CDCNET is installed, the version level changes. The version level is included in all released CDCNET software files except site-controlled configuration files. Version levels in the file names allow multiple versions of CDCNET to reside together on disk under one user name.

The version level of the components of CDCNET are controlled in two places: in the INITDCN procedure (stored in file INITDCN on the network administration user name) and in the exception list (file ELIST on the network administration user name). The exception list maintains the default version level for all non-host-connected DIs; INITDCN maintains the levels for all other components.

To update the version level, use the SETVL (SET_VERSION_LEVEL) procedure. Execute this procedure from the network administration user name:

```
BEGIN, SETVL, DCNPLIB, TOOL=tool, V=vvvv, ID=id.
```

Parameter	Description
TOOL=tool	The name of the component to update. Allowable values are: ANACD, ELIST, INMD, MANCC, NETOU, NPA, and ALL. INMD refers to the boot file level INITMDI loads into all host-connected DIs, ELIST refers to the exception list, and ALL updates all the tools.
V=vvvv	The version level of CDCNET you wish to activate.
ID=id	The 1-character identifier for the CHA1 template file. The id character for the release template is A. This parameter is required for TOOL=NETOU and TOOL=ALL.

It is recommended that TOOL=ALL be used for an upgrade or BCU installation. Once SETVL has completed, all future load requests for DIs and all accesses for ANACD, MANCC, and NPA use vvvv level software; the next initiation of NETOU uses template file TF_id_vvvv.

NOTE

For this procedure to properly process the exception list, the permanent file name must be ELIST, and the DEFBD (DEFINE_BOOT_DEFAULTS) command must specify the software version level with the text DV = vvvv(16). DV=vvvv(16) must be on one line. SETVL does not set the boot default for specific DIs that are handled in the exception list.

Installing a CDCNET Separate TIP

If you are installing CDCNET for the first time as part of a component order, follow the instructions in chapter 4. However, if you are adding a CDCNET separate Terminal Interface Program (TIP) in a component order, use the following instructions rather than the instructions in chapter 4:

1. Log in to the installation user name from an interactive terminal.

- Execute the following command:

```
RECLAIM, Z./DELETE, UN=NS2psrin, PF=*/PFGNDL1, PFGCHA1, PFGCHA2, PFGDCNS
```

This command deletes these files from the database. (The files are still there, but they are not processed during normal operations.)

- Update the RECLAIM database with information about the files on the component order tape by entering:

```
SYSGEN, UPGRADE, 0, 0, vsn, density.
```

Replace vsn with the VSN of the CDC supplied component order tape. (The VSN is listed in the media number field of the external tape label.) Replace density with the tape density of the permanent file tape (HY, HD, PE, or GE).

- Log in to the network administration user name from an interactive terminal.
- Execute the following command:

```
BEGIN, UPGRADE, DCNPLIB, netadun.
```

This command requests the permanent file tape and updates the files on the network administration user name. Since the CDCNET version level is the same as the original release level of CDCNET, the existing files whose names contain the CDCNET version level are overwritten. The only file modified is the CDCNET object library file L70vvvv, where vvvv represents the CDCNET version level.

Installation of the software is complete. You must now alter your DI configuration procedures to configure in the separate TIP. Consult the CDCNET Configuration Guide for details.

Installing CDCNET for MDI Reset Support Only

It may be desirable in a multihost network to manage all CDCNET software from a single mainframe. In this environment, CDCNET should be configured so that all DIs are loaded from software residing on one mainframe. To support the failure management software in the MDI, a small set of CDCNET software is required on the other mainframes which will not be providing load support. Specifically, each of the NOS hosts must be able to run the INITMDI software which is contained in the CHA1 portion of the CDCNET product. Installing this minimal portion of CDCNET results in a considerable saving in disk space. The instructions that follow describe how to install this minimal set of CDCNET software. Installation of a complete set of CDCNET software is described earlier in this section.

To execute these instructions, you must load the RECLAIM database to the installation user name and create the network administration user name. Do this by performing steps 1 and 2 in either chapter 3 or 4.

NOTE

If you have CDCNET installed on your system and want to change it to support only MDI resets, do not execute these instructions. Instead, purge all files on the network administration user name, except for DCNPLIB, then execute the command BEGIN, MDIONLY, DCNPLIB.

Enter these commands from the system console.

1. Enter:

```
X.DIS.  
USER,netadun,password.
```

Replace netadun with the network administration user name and password with its password.

2. Load the CDCNET permanent files from the permanent file tapes:

```
SYSGEN,RECLAIM, PFGDCNS.
```

3. Install file INITDCN, which is required by the INITMDI program, on the network administration user name:

```
BEGIN,MDIONLY, PFGDCNS.  
DROP.
```

4. During Step 5 of chapter 3 or 4, disable the loading of files PFGDCNS and PFGCHA2. File PFGCHA1 must be loaded to properly update the NAMSTRT file to contain the record JOBINMD and its associated JOB statement in the PARAM NAMSTRT records. This update is required to allow INITMDI to execute. Note that the installation of PFGCHA1 causes a number of other job records (NETLS, NETFS, and NETOU) to be added to NAMSTRT. These jobs are submitted when NAM is invoked, but since these applications are request startable, they will not execute at a control point or be in the rollout queue unless a load request is initiated for them by entries in the DI configuration file.
5. During Step 7 of chapter 3 or 4, there is no need to activate the new level of CDCNET since there is no CDCNET software to load.

Installation Wrapup Activities

After you have upgraded CDCNET and the new version is running satisfactorily, remove earlier versions of the CDCNET software from disk to conserve disk space. The CLEANUP procedure purges software files from the network administration user name. Only files from the specified version level are removed. Boot files and object libraries are also removed from the network directory file NETDIR.

To remove a version of CDCNET software, execute the following procedure call from the network administration user name:

```
BEGIN,CLEANUP,DCNPLIB,V=vvvv.
```

Replace vvvv with the version level of the CDCNET files you want to remove from disk.

NPA Database Maintenance

When CDCNET is first installed, the NPA databases are initialized to empty direct access permanent files on the network administration user name. By running the REFCLF (REFORMAT_CDCNET_LOG_FILE) program described in the NPA Reference Manual, these databases are updated from the log files stored on user name SYSTEMX. The databases are manipulated and reports are generated by NPA Report which is also documented in the NPA Reference Manual. DCNPLIB contains procedure DB, which

performs several functions on the databases and NPA database definition files. The procedure calls are described next.

- **BEGIN(DB,DCNPLIB,DEFINE).** This procedure call creates the NPA database files if they do not already exist. Existing database files are not altered. This procedure does not affect database definition files (which are installed as part of NPA).
- **BEGIN(DB,DCNPLIB,PURGE).** This procedure call purges the NPA databases. The database definition files are not affected.
- **BEGIN(DB,DCNPLIB,PERMIT,username).** This procedure call permits the specified user name to have read access to NPA databases and database definition files. Permits are controlled with the standard NOS PERMIT command. The parameter username is any valid NOS user name.
- **BEGIN(DB,DCNPLIB,DEPERMIT,username).** This procedure call prevents the specified user name from accessing NPA databases and database definition files by removing it from the permit list for each database. Permits are controlled with the standard NOS PERMIT command. The parameter username is any valid NOS user name.

CDCNET Host Application (CHA1) Product Content

The CDCNET host application procedure consists of the following components and utilities:

- HOST message template file
- INITMDI (Initialize MDI)
- NAMSTRT records
- NETBDF (Build CDCNET Directory File)
- NETFM (Network File Manager)
- NETFS (Network File Server)
- NETITF (Initialize NETOU Template File)
- NETLS (Network Log Server)
- NETMDF (Merge CDCNET Directory File)
- NETOU (Network Operator Utility)
- NETPLM (Network Procedure Library Manager)
- NETRDF (Restructure CDCNET Directory File)
- NLTERM (Network Log File Termination Utility)
- PIM (MDI Loader PP Program)

CDCNET requires NAM and many of its programs and procedures.

The CDCNET network applications installation procedure modifies the network startup master file (NAMSTRT), adds new job startup calls into the network startup records (INIT, MULTI, etc.) and adds the corresponding job skeletons.

Entries for the DIs must be specified in the EQPDECK before a CDCNET network can be functional. This information is in the NOS Version 2 Analysis Handbook.

The CDCNET Network Operations and Analysis manual and the CDCNET Configuration Guide describe the CDCNET Host Applications and Utilities.

Installing CDCNET rebuilds the NAM network file collector. The collector is modified to collect the many dump and trace files that the CDCNET Host Applications and Utilities generate.

Unique Parameters

Parameter	Description
DEBUG	To add code to aid in debugging and maintenance, specify the keyword DEBUG on the call to the CHA1 installation procedure. NETFS, NETLS, NETOU, and NLTERM abort on certain error conditions.
NOTRACE	To disable AIP trace file creation for the CDCNET Host Applications and Utilities, specify the keyword NOTRACE on the call to the CHA1 installation procedure.
ID	This parameter sets the identifying character (A..Z,0..9) for the template file, HTF_id_psout. The default is A. Change this character whenever CHA1 is rebuilt with a change that affects the template file. This parameter allows you to have multiple versions of the host template file. This id parameter is also supplied to the GENMSG procedure for creating the combined template file.

USER File Directives

To disable usage of SORT5 by NETFM, include the directive:

```
*DEFINE NOSRT5
```

on a USER file when executing the CHA1 installation procedure. By specifying this directive, NETFM does not use SORT5 and NETFM list output is displayed in unsorted order. Use of this directive is not recommended.

Installation Parameters

The following CHA1 installation parameters are defined in the common deck INPARU. To change these parameters, place the appropriate Update directives in a USER file and apply the file to the CHA1 installation procedure.

Parameter	Default	Description
NFA\$NFILES	50	Number of CDCNET file entries the directory is expected to contain. This parameter is used by the NETBDF utility to build a directory file (NETDIR).
NFA\$PUN	NETOPS	Primary user name that is allowed WRITE permission on the directory file.
NFS\$ALTUN	NETADMN	The user name that is permitted WRITE access to files created by the Network File Server. Alternate catlist of these files is also available to this user name.
NLS\$ALTUN	NETADMN	The user name that is permitted READ APPEND (RA) access to created log files. Alternate catlist of the created log files is available to this user name.

The following CHA1 installation parameter is defined in the common deck NETCOM. To change this parameter, place the appropriate Update directive in a USER file and apply the file to the CHA1 installation procedure.

Parameter	Default	Description
NFA\$CATALOG	NETDIR	Name of the CDCNET directory file.

The deck INPARU also contains installation parameters that enable the site to select retain timer values pertinent to Network File Server (NETFS) performance. The retain timer parameters and their default values are listed below.

Parameter	Default Retain Timer Value (in seconds)	CDCNET File Type
RTBOOT	5	BOOT
RTCONFIGUR	5	CONFIGURATION
RTCONFPLIB	30	CONFIGURATION LIBRARY
RTDUMP	0	DUMP
RTEXCEPTION	5	EXCEPTION
RTLIBRARY	180	MODULE LIBRARY
RTLOAD	30	LOAD_PROCEDURE
RTLOADPLIB	30	LOAD_PROCEDURE LIBRARY
RTTERMINAL	30	TERMINAL_DEFINITION_PROCEDURE
RTTERMPLIB	30	TERMINAL_DEFINITION_PROCEDURE LIBRARY
RTUNDEF	0	All others
RTUSER	30	TERMINAL_USER_PROCEDURE
RTUSERPLIB	30	TERMINAL_USER_PROCEDURE LIBRARY

Message Templates

When CHA1 is built, it creates a message template file, HTF_id_psrou, in addition to binaries for the deadstart tape and NAMSTRRT records. In the file, id is a single character supplied to the CHA1 installation procedure, and psrou is a common DECKOPL parameter set in COMMOD. The template file created by CHA1 is combined with the template file released by CDCNET, CTF_vvvv where vvvv is the CDCNET version level. These files are combined using the GENMSG procedure in file DCNPLIB on the network administration user name to create a combined binary template file, TF_id_vvvv. GENMSG should be executed whenever CHA1 is rebuilt or a CDCNET BCU is installed. Call the GENMSG procedure from the network administration user name.

```
BEGIN(GENMSG,DCNPLIB,ID=id,V=vvvv,P=psrou[,PURGE])
```

Parameter	Description
ID=id	The id character of the template file created by CHA1. HTF_id_psrou is attached from the installation user name. If the file is not found, GENMSG ATTACHes the file from the network administration user name.
P=psrou	The PSR level of CHA1.
PURGE	A keyword to tell GENMSG to PURGE the combined template file TF_id_vvvv if it exists. Without this parameter, GENMSG by default does not overwrite an existing TF_id_vvvv file.
V=vvvv	The CDCNET version level to combine with your host template file. CTF_vvvv is ATTACHed from the network administration user name.

Once GENMSG is complete, you can activate the new file using the SETVL procedure described earlier in this section. Make sure you specify NETOU for the tool parameter when executing SETVL. The next invocation of NETOU will use the new template file.

NAM Startup Procedure File Changes

The CDCNET network is started using the same commands (NAMNOGO, etc.) as the CCP network. Refer to the NAM Procedure File subsection in the NAM5 section of this chapter.

CDCNET applications use the same NAMI job startup procedure as does NAM, with the following additions to NAMSTRT:

- Application startup calls in network startup records:

Call	Description
JOB(JOBFS,SF,SY,NS)	Network File Server
JOB(JOBOU,UO,SY,NS)	Network Operator Utility
JOB(JOBL,SL,SY,NS)	Network Log Server
JOB(JOBINMD,IN,SY,NS)	INITMDI (MDI Initializer)
JOB(JOBFSR,FS,DI,SY,NS)	File Server Restart Job
JOB(JOBLSR,LS,DI,SY,NS)	Log Server Restart Job
JOB(JOBOUR,OU,DI,SY,NS)	Operator Utility Restart Job

- Job skeleton records JOBFS, JOBOU, JOBL, JOBFSR, JOBLSR, JOBOUR, JOBINMD
- Parameters for the standard NAM startup records:

Parameters	Description
PARAM(CDCNET=YES)	Informs collector job, JOBCOL, to process dump files.
PARAM(FSSTART=NO)	Prevents the Network File Server from starting when NAM comes up.
PARAM(LSSTART=NO)	Prevents the Network Log Server from starting when NAM comes up.
PARAM(OUSTART=NO)	Prevents the Network Operator Utility from starting when NAM comes up.
PARAM(ZZFC=500)	Flush count for log file buffer (used only by NETLS).

NOTE

A YES value on any of the middle three parameters listed causes that application to start when NAM comes up. When NO is used, the applications start automatically when they are needed.

CDCNET Network Host Application Program Requirements

Job skeleton records for NETFS, NETLS, NETOU, and INITMDI jobs must contain a command that calls the program the job intends to run. These commands have the following format:

```
prog(parameter1,...,parametern)
```

Field	Description
prog	Program call. May be NETFS, NETLS, NETOU, or INITMDI.
parameter _i	Unless specified otherwise, these are optional, order-independent parameters. They may be of the form: keyword=value or keyword

The files used by each program are listed following the description of each program call.

NETFS - Network File Server

NETFS requires the following command:

```
NETFS (MC=mc , NIN=nin)
```

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=0. The default value, which is 500, may be reset using the ZZMC PARAM statement.
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAM when the network operation is started. This parameter is required.

NETFS requires the following files:

File	Description
NETDIR	The network directory contains entries that match NOS permanent files to logical file references used by CDCNET Host Applications and Utilities. A default NETDIR is provided with the network.
NRF1	The job record to be copied to the ZZZZZDN file by NETREL. This job record determines the disposition of the network trace file when NETREL is called on a periodic basis.
NRF2	The job record to be copied to the ZZZZZDN file by NETREL. This job record determines the disposition of the network trace file when NETREL is called as a result of an operator command.

NOTE

The default job skeleton of NETFS creates NRF1 and NRF2 from the input records of the NETFS job.

NETLS - Network Log Server

NETLS requires the following command:

NETLS (MC=mc ,NIN=nin,FC=fc)

Parameter	Description
FC=fc	Flush count; specifies the number of characters received in log data before the log buffer is flushed and an end-of-record is written. The default, which is 2750 characters, may be reset using the ZZFC PARAM statement. If the value is one, the buffer is flushed after each log block.
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=0. The default value, which is 500, may be reset using the ZZMC PARAM statement.
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAM when the network operation is started. This parameter is required.

NETLS requires the following files:

File	Description
NRF1	The job record to be copied to the ZZZZZDN file by NETREL. This job record determines the disposition of the network trace file when NETREL is called on a periodic basis.
NRF2	The job record to be copied to the ZZZZZDN file by NETREL. This job record determines the disposition of the network trace file when NETREL is called as a result of an operator command.

NOTE

The default job skeleton of NETLS creates NRF1 and NRF2 from the input records of the NETLS job.

NETOU - Network Operator Utility

NETOU requires the following command:

```
NETOU (NIN=nin, MC=mc, DM=dm, PROMPT, TDF=lfm)
```

Parameter	Description
DM=dm	Node number of the MDI to be used as the default. If you select a specific MDI, this MDI is used to communicate with CDCNET. If the parameter is not specified or the operator software in the MDI is not connected to the host, the MDI that has been connected the longest is used as the default MDI.
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=0. The default value, which is 500, may be reset using the ZZMC PARAM statement. If NETOU was built with the NOTRACE option, then no messages are written.
NIN=nin	Network invocation number. One- to three-digit decimal number assigned by NAM when the network operation is started. This parameter is required.
PROMPT	If the parameter is present and there is more than one MDI connected to NETOU, all the operators are prompted for selection of an MDI when the operator logs in. If this parameter is not present, the default MDI is transparently selected for an operator at login.
TDF=lfm	The local file name of the template definition file. This file contains the templates used to format messages for display and must have been generated by the utility NETITF. The default local file name is TEMPLTB .

NETOU requires the following files:

File	Description
INITDCN	Procedure indicating parameter values for id and vvvv for template file name. Must be stored on the network administration user name. This file is created as part of the CDCNET installation.
NRF1	The job record to be copied to the ZZZZZDN file by NETREL. This job record determines the disposition of the network trace file when NETREL is called on a periodic basis.
NRF2	The job record to be copied to the ZZZZZDN file by NETREL. This job record determines the disposition of the network trace file when NETREL is called as a result of an operator command.
TF_id_vvvv	The default file name of the template definition file. This file contains the templates used to format messages for display and must have been generated by the utility NETITF. The file name being used is controlled by using the INITDCN procedure on the network administration user name.

NOTE

The default job skeleton of NETOU creates NRF1 and NRF2 from the input records of the NETOU job.

INITMDI - Initialize MDI

INITMDI requires the following command:

```
INITMDI (L=lfm1 , NAM=nam, DT=dt , NIN=nin)
```

Parameter	Description
DT=dt	Device type (EST name). This parameter is required and consists of two alphanumeric characters. Refer to the NOS Version 2 Analysis Handbook for more information. For an MDI, this parameter is DT=ND.
L=lfm1	Message history file. If L=lfm1 is not specified or L=0 is specified, no message file is produced. Messages concerning load, dump, and errors are written to this file. L=OUTPUT should be used.
NAM=nam	If this parameter is not specified or NAM=NO is specified, the NAM interface is assumed to not be present. If NAM or NAM=YES is specified, the NAM operator interface is used for error messages.
NIN=nin	Network invocation number: 1- to 3-digit decimal number that is placed in the NIN field of messages written into the L file. If not specified, the default is 000.

INITMDI uses the NOS Exception List in the network directory (NETDIR) to determine which boot file to use to load the MDI; therefore, the following parameters, formerly used by INITMDI, have either become redundant or invalid:

```
B=lfm
D=xx
EST=est
V=ver
```

If an INITMDI call contains any of these parameters, the values are ignored and a dayfile message is displayed. All boot files to be used should be correctly defined in NETDIR; for example, NF=BOOT#vvvv_MCI where vvvv is the 4-digit hexadecimal version number.

The actual call to INITMDI is contained in the procedure INITDCN on the network administration user name. Any parameter changes to INITMDI should be made in that file. The released INITMDI call is of the form:

```
INITMDI (L=OUTPUT, DT=ND, NIN=CIN) .
```

INITMDI requires the following files:

File	Description
NETDIR	The network directory contains entries that match NOS permanent files to logical file references used by CDCNET Host Applications and Utilities. A default NETDIR is provided with the network.
INITDCN	This procedure contains the actual INITMDI call. The primary function of INITDCN is to set the version level (V=ver) parameter. This file is created as part of the CDCNET installation.

Network Definition Language (NDL) Requirements

To run CDCNET, you must modify the Local Configuration section of the existing Network Definition File to include application definition (APPL) statements and INCALL statements. The released NDL file NDLDATA, on the network administration user name, contains an example of the required statements. They are as follows:

```

LCFFILE      :LFILE.
NETOU        :APPL KDSP,RS,PRIV.
NETFS        :APPL KDSP,RS,PRIV.
NETLS        :APPL RS,PRIV.
NLTERM       :APPL KDSP,PRIV.

INCALL       FAM=0,UNAME=NETOPS,ANAME=NETOU,
              DBL=7,ABL=7,DBZ=2043,UBZ=20,UBL=7.
INCALL       FAM=0,UNAME=NETOPS,ANAME=NETLS,
              DBL=7,ABL=7,DBZ=2043,UBZ=20,UBL=7.
INCALL       FAM=0,UNAME=NETOPS,ANAME=NETFS,
              DBL=7,ABL=7,DBZ=2043,UBZ=20,UBL=7.
INCALL       FAM=0,UNAME=NETOPS,ANAME=NLTERM,
              DBL=7,ABL=7,DBZ=2043,UBZ=20,UBL=7.

END.
    
```

NOTE

The user name NETOPS, in the previous text, refers to the Control Data default value for the network operations user name. If you have changed this value, substitute your site name for the Control Data default value.

For complete information on the meaning of the statements, refer to the Network Definition Language Reference Manual.

CDCS2 - CYBER Database Control System Version 2

This section describes these installation options for CDCS2:

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

Unique Parameters

Parameter	Description
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 the COBOL5 section in this chapter.

CDC Procedure File

Refer to Subsystem Initiation, at the beginning of this chapter, for information about the CDC startup procedure file and subsystem initiation. Refer to the CYBER Database Control System 2 Reference Manual for instructions on constructing the procedure file.

Special Notes

- CDCS2 users must have permission to use the system control point facility and the system control point facility must be enabled. Refer to the NOS Version 2 Administration Handbook for a description of MODVAL and the NOS Version 2 Analysis Handbook for information on the SCP IPRDECK entry.
- To activate a debug trace facility for CDCS2, 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. |
| | DFASK Ask restart identifier |
| | DFBEG Begin transaction |
| | DFCLS Close |
| | DFCMT Commit transaction |
| | DFDBS Database status block |
| | DFDEL Delete |
| | DFDRP Drop transaction |
| | DFEND End |
| | DFGID Get restart identifier |
| | DFINV Invoke |
| | DFLOG Logging |
| | DFLOK Lock |
| | DFOPN Open |
| | DFPVC Privacy |
| | DFRD1 Sequential read |
| | DFRD2 Random read |
| | DFREW Rewrite |
| | DFRPT Recover point |
| | DFRSR Relation start |
| | DFRWF Rewind area file |
| | DFRWR Rewind relation |
| | DFRWX Rewind index file |
| | DFRX1 Read sequential on index file |
| | DFRX2 Read random on index file |
| | DFSKF Skip |
| | DFSTR Start |
| | DFSTX Start on index file |
| | DFTER Abnormal termination |
| | DFULK Unlock |
| | DFVER Version change |
| | DFWR2 Random write |
| 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
IS	IS primary key charge
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 microseconds, and the I/O charge is 7000 microseconds.

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 decks to obtain sequence information.

Parameter	Default	Description
BREAKTABSIZ E	16	Number of entries in breakpoint table. Parameters are located in common decks BREAKD (SYMPL) and BREAKZ (COMPASS).
TABSIZ	16	
GROUPTABSIZ E	16	Number of entries in group table. Parameters are located in common decks GROUPD (SYMPL) and GROUPZ (COMPASS).
TABSIZ	16	
TRAPTABSIZ E	16	Number of entries in trap table. TRAPXSIZE and XSIZE must each be three greater than the tablesize defined by TRAPTABSIZ and TABSIZE. Parameters are located in common decks TRAPD (SYMPL) and TRAPZ (COMPASS).
TRAPXSIZ E	19	
TABSIZ	16	
	19	
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.

CML - Concurrent Maintenance Library

There is no installation procedure for CML in the DECKOPL procedure file. Refer to the Concurrent Maintenance Library Reference manual for installation information.

COBOL5 and COBOL5Q - COBOL Version 5

This section describes the following installation options for COBOL5:

- 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 (or quick) mode installation (deck COBOL5Q).

The Q mode installation procedure option allows you to build a COBOL5 compiler that has been modified. It compiles or assembles only those routines which have been modified and combines the binaries produced with a previous release level of COBOL5 to create a new compiler by use of the COPYL utility. You must provide *COMPILE directives on a USER file for any affected routine.

The purpose of using the Q mode procedure is to allow you to modify the COBOL5 compiler without compiling and assembling the entire compiler. This mode should be used only to modify the compiler with corrective code or user code which does not affect common decks. For example, you may want to use the Q mode installation option to activate Data Management, set default page size, or enable COBOL5 to use CMU.

Unique Parameters

The COBOL 5 compiler is released supporting the TAF and CDCS interfaces. The compiler will function correctly if you do not install either TAF or CDCS. By default, the compiler will be built supporting the interfaces. To disable either or both interfaces, use the keyword parameters below.

Parameter Description

NOCDCS	To deactivate the interface between COBOL5 and CDCS2, specify the keyword NOCDCS on the call that executes the installation procedure for COBOL5 or COBOL5Q. Relocatable binaries created by compiling COBOL5 programs on a system that has the COBOL5 CDCS interface do not execute on a system that does not have this interface. The reverse is also true: relocatable binaries created by compiling on a system that does not have the COBOL5 CDCS interface fail to execute on a system that does have the CDCS interface.
NOTAF	To deactivate the interface between COBOL5 and TAF, specify the keyword NOTAF on the call that executes the installation procedure for COBOL5 or COBOL5Q.

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 a USER file 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 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 2 Version 2.0

This section describes the following:

- Product Dependencies
- SYSGEN Functions

Product Dependencies

The COBOL5 compiler and library must be available to install DCL. The product must run from permanent files.

SYSGEN Functions

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

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

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

2. Run the DCL installation procedure. When the procedure is finished executing, enter these commands at the system console:

```
X.DIS.
USER (SYSTEMX, password)
ATTACH (PFGDCL2/UN=insun)
SYSGEN (DCL2)
```

DUAL - Dual State

This section describes the following:

- Unique Parameters
- Configuration Information
- Dual State Source Library
- Dual State Upgrade of NOS Only
- Dual State on Older NOS Systems
- NOS and NOS/VE Memory Allocation
- Allocation of Dual State Peripheral Processors
- Dual State Minimum Hardware Requirements
- NOS/VE Initiation
- VEIAF Family Name Selection
- NOS SETVE Modification
- NOS RUNJOBS Modification
- NOS ACCFILE
- NOS ACCFILE Modification

Unique Parameters

Parameter	Description
TRACE	To enable AIP tracing, specify the keyword TRACE on the DUAL installation procedure.
NOSLEV	Can be three numeric characters. Denotes the level of NOS used to build dual state. In the released binaries, this was set to psrout.
VELEV	Can be five alphanumeric characters. Denotes the level of NOS/VE used to build dual state. In the released binaries, this was set to psrout. In a Batch Critical Update, it was set to the correction level.
CSET	Denotes the character set of the system. Valid options are 64 or 63. The default character set is 64.

NOTE

Parameters NOSLEV and VELEV are used for comment purposes only (binary IIAPAS). This enables sites to easily determine the operating system levels used to create the dual state binaries. This is important when reporting problems or keeping track of dual state BCUs.

Configuration Information

To configure a dual state system, create the following:

- NVE user name. Create user name NVE for running the NVE subsystem. (The user name is created automatically with the password of NVEX in a first-time installation.) If you create the user name yourself, the user name should have these additional validations added to the default NOS validations:

```
AP=IAF, AP=VEIAF, AW=CASF, AW=CAND, AW=CCNR, AW=CLPF, AW=CSPF,
AW=CUCP, AW=CNVE, AW=CTIM, AW=CUST, AW=CQLK, CC=77B, CM=30B,
CS=7, DB=2, DF=77B, FS=6, LP=77B, MS=76B, MT=1, SAV=CULT,
SL=77B, TL=77B, VM=CT
```

- LIDCMid file. This file defines the physical and logical machines available to your system. (The id in LIDCMid is the two alphanumeric characters from the MID CMRDECK entry that make up your machine identifier. The released default id is AA.) The number of entries in the LIDCMid file determine the value specified in the LDT CMRDECK entry. The LIDCMid file is stored as an indirect access permanent file on user name SYSTEMX (user index 377777B). In addition to entries in this file for dual state, entries are also made for PTF/QTF, the Remote Host Facility (RHF), and the NOS Scope Station Facility. A sample LIDCMid file is provided with the release materials. This file should be edited on the installation user name, then moved to user name SYSTEMX.

For a first-time installation, GET the sample file from user name SYSTEMX directly. If you are installing dual state for the first time during a NOS upgrade, execute the SYSGEN(DUAL,LID) procedure from the installation user name to load a copy of LIDCMid (and LIDVEid) to the installation user name.

To move the LIDCMid file to user name SYSTEMX, execute the command SYSGEN(MOVE,LIDCMid,SYSTEMX,,Y,PERMIT) from the system console when directed to by instructions in chapters 2, 3, or 4. (The PERMIT parameter allows read access to the file from the installation user name.) Once the file has been moved, build the LID table in central memory using the CLDT system program. To use CLDT, ensure that NAM, IAF, RHF, and SSF are not running and enter the command X.CLDT. from the system console. (CLDT is executed automatically at NOS deadstart if an LDT entry exists in the CMRDECK.)

- LIDVEid file. This file defines the Logical Identifiers (LIDs) used by NOS/VE batch jobs. (The id in LIDVEid is the two alphanumeric characters from the MID CMRDECK entry that make up your machine identifier. The released default id is AA.) LIDVEid is an indirect access permanent file stored on user name SYSTEMX (user index 377777B). A sample LIDVEid file is provided with the release materials. This file should be edited on the installation user name, then moved to user name SYSTEMX.

For a first-time installation, GET the sample file from user name SYSTEMX directly. If you are installing dual state for the first time during a NOS upgrade, execute the SYSGEN(DUAL,LID) procedure from the installation user name to load a copy of LIDVEid (and LIDCMid) to the installation user name.

To move the LIDVEid file to user name SYSTEMX, execute the command X.SYSGEN(MOVE,LIDVEid,SYSTEMX,,Y,PERMIT) from the system console.

- LDT CMRDECK entry. This entry defines the size of the Logical Identifier Table in central memory. The size of the table is determined by the number of entries in the LIDCMid file.
- VE CMRDECK entry. This entry in combination with or instead of the MINCM CMRDECK entry defines NOS/VE and the amount of memory to reserve for NOS/VE. Refer to NOS and NOS/VE Memory Allocation later in this section.
- MINCM CMRDECK entry. This entry defines the amount of memory to reserve for NOS. Refer to NOS and NOS/VE Memory Allocation later in this section.
- EQPDECK entries for sharing memory. The VEMEM utility, documented under NOS and NOS/VE Memory Allocation later in this section, helps determine the entries needed to allocate memory between NOS and NOS/VE.
- EQPDECK entries for sharing peripherals. This topic is discussed in the Deadstart Decks section of the NOS Version 2 Analysis Handbook.
- IPRDECK entries for controlling initiation of NOS/VE. See NOS/VE Initiation later in this section.

For more information about the LIDCMid and LIDVEid files and the LDT CMRDECK entry, refer to the LID/RHF Configuration Files section of the NOS Version 2 Analysis Handbook. For additional information about the CMRDECK and EQPDECK entries, refer to the Deadstart Decks section of that handbook.

Dual State Source Library

The Dual State Source Library is in multifile PFGDUAL (for BCUs, it is in multifile PFGBCU2). To access it, execute the following command from your installation user name:

```
SYSGEN (DUAL, SOURCE)
```

For BCUs, execute:

```
SYSGEN (DUAL, SOURCE, BCU)
```

This creates (or replaces) the file named VEDSSL, which contains the Dual State Source Library in NOS/VE format.

To rebuild the dual state components or to modify the software, you must transfer file VEDSSL to NOS/VE. To modify the file on NOS/VE, you must use the NOS/VE Source Code Utility. Once all desired modifications (if any) are complete, the source library is converted to a text file that is transferred back to NOS. The text file is used as input to the NOS DECKOPL installation procedure DUAL for compiling the dual state components. The dual state source library should be kept on a NOS/VE user name from which other NOS/VE maintenance activities can be performed.

This user name should give read execute permission to the file

```
$SYSTEM.SOFTWARE_MAINTENANCE.RAF$LIBRARY
```

This can be done from the NOS/VE console with the command:

```
CREATE_FILE_PERMIT $SYSTEM.SOFTWARE_MAINTENANCE.RAF$LIBRARY ..
G=USER FN=familyname U=username
```

where familyname and username are for the analyst who maintains the Dual State Source Library.

1. To transfer the library to NOS/VE, log in to NOS/VE and execute these commands:

```
CHANGE_LINK_ATTRIBUTES U=(insun,nosfamily) PW=batchpassword
GET_FILE $USER.DUAL_STATE_SOURCE_LIBRARY VEDSSL DC=B56
```

2. To transfer the modified source library to NOS, use these commands:

```
CREATE_COMMAND_LIST_ENTRY E=$SYSTEM.SOFTWARE_MAINTENANCE.RAF$LIBRARY
CONVERT_DSSL_TO_TEXT $USER.DUAL_STATE_SOURCE_LIBRARY DUALpsrin
CHANGE_LINK_ATTRIBUTES U=(insun,nosfamily) PW=batchpassword
REPLACE_FILE DUALpsrin
```

Replace psrin with the PSR level of NOS. This creates a file named DUALpsrin on the NOS installation user name. When the DUAL installation procedure is executed, it picks up this file and uses it as input.

NOTE

The dual state installation procedure (DUAL) must always match the level of the Dual State Source Library being assembled. For example, to assemble NOS/VE L664 with a NOS L647 system, you must use the L664 DECKOPL installation procedure for dual state.

Dual State Upgrade of NOS Only

Certain back levels of NOS/VE are supported on the released level of NOS. These are determined by the support window maintained by Control Data (refer to the NOS/VE Installation Handbook). To create the binaries that support these systems, you must assemble the dual state product. To assemble dual state for these combinations, the following items are required:

- The dual state source and corresponding DECKOPL installation procedures from the desired level of NOS/VE
- GLOBLIB, PRODUCT, and other needed source from the desired level of NOS

NOTE

If other products need to be customized during this upgrade to the new NOS level, you should assemble them first by following the instructions in chapter 6, Customizing Installations. This ensures that the correct output PLs are available when customizing dual state. After completing the steps in chapter 6, return here to assemble dual state.

After setting the installation procedures to specify the desired NOS level and SEEDing with the NOS deadstart tape, dual state can be reassembled. This is accomplished by following the example below. Similar steps are performed for other combinations.

You are running a NOS L678–NOS/VE L678 production system. You have received L688 materials, but wish to upgrade NOS only. As usual, you start executing the instructions in chapter 3, Upgrade Installation. During Step 2, you will be instructed to refer to chapter 6. Chapter 6 mentions some notes and refers you to this section.

The following are the files and their PSR levels that are required to assemble binaries to support a NOS L688–NOS/VE L678 system.

L688	L678
OPLpsrout	DUAL678
TDU1688	DECKOPL
BAM1psrout	INSTALL
	COMMOD

1. Perform Step 1 of chapter 6. This procedure loads the L688 RECLAIM database and various other files that are needed. The L688 release NOS permanent file and deadstart tapes are needed. Use the L688 release deadstart tape (or a L688 customized NOS deadstart tape, if any) as file SYSTEM.
2. Get the L678 release NOS deadstart and permanent file tapes. If you have applied a L678 BCU to dual state, you will also need that tape.
3. Log in to the installation user name and use RECLAIM to obtain the L678 installation procedures.

```
RECLAIM, DB=0, Z./COPY, TN=vsn, D=density, FN=PFGEDECK
```

Replace vsn with the VSN of the L678 CDC-supplied permanent file tape. The VSN is in the media number field of the external tape label. Replace density with the density of the permanent file tape (HY, HD, PE, or GE).

4. Use RECLAIM to obtain the L678 Dual State Source Library.

```
RECLAIM, DB=0, Z./COPY, TN=vsn, D=density, FN=filename
```

If you have applied a BCU to dual state, replace filename with PFGBCU2 and vsn with the VSN of the L678 BCU tape. If no BCU has been applied, replace filename with PFGDUAL and vsn with the VSN of the L678 CDC-supplied permanent file tape. The VSN is in the media number field of the external tape label. Replace density with the density of the tape (HY, HD, PE, or GE).

5. If they exist, change the names of files DECKOPL, COMMOD, INSTALL, VEDSSL, and DUAL688 to something else. Use the local files that were RECLAIMed in the previous steps to create the installation procedure files and the Dual State Source Library.

```
REWIND, PFGDECK.
DEFINE, DECKOPL, INSTALL, VEDSSL.
COPYBF, PFGDECK, DECKOPL.
COPYBF, PFGDECK, INSTALL.
COPYBF, PFGDECK, COMMOD.
SAVE, COMMOD.
```

- If no BCU has been applied to NOS/VE, enter the following command:

```
REWIND, PFGDUAL.
COPYBF, PFGDUAL, VEDSSL.
```

- If a BCU has been applied to NOS/VE, enter the following command:

```
REWIND, PFGBCU2.
COPYBF, PFGBCU2, VEDSSL.
```

6. Return all the files and convert the Dual State Source Library to an input program library for the DUAL installation procedure.

```
RETURN, PFGDECK, PFGDUAL, PFGBCU2, VEDSSL, DECKOPL, COMMOD, INSTALL.
```

To convert the Dual State Source Library, follow the instructions in the Dual State Source Library subsection, earlier in this section. However, since file VEDSSL was created in step 5, skip the SYSGEN(DUAL,SOURCE) or SYSGEN(DUAL, SOURCE,BCU) command. When this step is complete, a file named DUAL688 will have been created.

7. Perform Step 2 of chapter 6, using the L678 DECKOPL materials just loaded. Be sure to modify COMMOD so that psrin is 688 and psrout matches the level of your output PLs. You must also add the following modification to COMMOD (note that BLDUN matches the psrin level).

```
*DECK COMBLUN
*D 1
BLDUN= (*N=NS2688),
```

Since older copies of the installation procedures are being used, it is not recommended to reassemble any product other than dual state.

8. Perform Step 3 of chapter 6, if you have user code for decks VER or 1VP.
9. Perform Step 4 of chapter 6. Be sure to use the L688 release deadstart tape (or the L688 customized NOS deadstart tape, if any) for the LOCAL file named SYSTEM when executing the SEED procedure.
10. Step 5 may be skipped, since the composite OPL would have been rebuilt as part of the customizing performed earlier.
11. Execute the DUAL installation procedure. (Refer to Step 6 and table 6-6 in chapter 6.)

12. Perform Step 7 of chapter 6 to create a new deadstart tape. If you already customized some L688 products and have that deadstart tape available, use that tape as the base deadstart tape for this step. If you have not customized any other products, use the release L688 deadstart tape as your base.
13. If you need to customize more products at a later time, files DECKOPL, INSTALL, and COMMOD need to be changed back to their L688 NOS versions. Refer to the files whose names were changed in step 5 of this section.

You now have a deadstart tape containing L688 NOS–L678 NOS/VE. You would return to Step 3 of chapter 3, as per previous instructions in Step 2 of that chapter.

Dual State on Older NOS Systems

The released level of NOS/VE is supported on certain backlevels of NOS. These are determined by the support window maintained by Control Data (refer to the NOS/VE Installation Handbook). The binaries are contained in RECLAIM dump format on file PFGDUAL. They are loaded to disk by executing the following command from an interactive terminal logged in to the installation user name. Remember that the RECLAIM database must be updated and the SYSGEN procedures must be current (refer to Step 2 in chapter 3).

```
SYSGEN, DUAL, OLDNOS.
```

This creates files with the naming convention NVEoldlev, where oldlev is the PSR level of NOS that was used. For example, NVE647 would be a file containing binaries to run the released level of NOS/VE on a NOS level 647 system.

To apply these binaries to your production system and create a new deadstart tape, use the following commands:

```
RETURN, SYSTEM.  
COMMON, SYSTEM.  
ATTACH, NVEoldlev.  
SYSGEN, DST, SYSTEM, NVEoldlev, NEW, 0, density.
```

Replace oldlev with the NOS level you are running and replace density with the tape density of the new deadstart tape.

NOS and NOS/VE Memory Allocation

When a mainframe is executing dual state, part of the memory is used by NOS/VE and part is used by NOS. Entries in the CMRDECK and EQPDECK determine how the physical memory is shared. A set of NOS procedures named VEMEM have been provided to help determine the deadstart deck entries needed. If file VEMEM does not exist on the installation user name, run SYSGEN(DUAL,LID) from the installation user name. This also creates files LIDVEid and LIDCMid.

To execute VEMEM, enter the following command from an interactive terminal logged in to the installation user name:

```
LINE .
BEGIN, VEMEM, VEMEM.
```

VEMEM asks you a series of questions about the physical memory of the mainframe, as well as how you want to divide the memory between NOS and NOS/VE. The utility also helps you allocate your NOS memory for user accessible extended memory (UEC), unified extended memory (UEM), and 895 input/output buffers (IOB).

VEMEM contains extensive help entries and validates all of your input. To obtain help during a prompt, enter a question mark (?) followed by a CR.

When VEMEM has completed, it displays your required CMRDECK and EQPDECK entries. The entries are also stored on local file CMREQP.

A more technical discussion of shared memory is given in the NOS Version 2 Analysis Handbook.

Allocation of Dual State Peripheral Processors

When running in a dual state environment, there are some instances when NOS/VE is not able to obtain additional peripheral processors from its CYBER 170 host. This behavior is partially explained by the method in which NOS allocates peripheral processors to NOS/VE and the type of peripheral processors NOS/VE is requesting.

The following rules are used to allocate peripheral processors to NOS/VE:

1. The CYBER 170 operating system must have a minimum of four pool peripheral processors available for its own use. The four pool peripheral processors do not include dedicated peripheral processors which contain MTR, DSD, SCI and similar programs. If the current request would cause the minimum to fall below four, the request is not granted.
2. If the mainframe is a CYBER 810 or 830 with 20 peripheral processors, there must be at least 2 driver peripheral processors available for use by NOS. Driver peripheral processors are numbered 20 through 31 on NOS, and 2 through 11 on NOS/BE. The numbers are octal on each system. Dedicated peripheral processors are excluded from this count.

The minimum number of pool and driver peripheral processors is determined by assembly constants in common deck COMSVED. The minimum number of pool peripheral processors is named MINP and the minimum number of driver peripheral processors is named MINDP. The released values for these parameters are 4 for MINP and 2 for MINDP (the minimum values to which these parameters should be set). For more information, refer to the description of COMSVED described in the NOS section of this chapter.

For performance reasons, NOS/VE requests a driver peripheral processor for disk, tape, and other drivers. If a driver peripheral processor is unavailable after 10 seconds, NOS/VE requests any available peripheral processors. Performance may be degraded in this situation but it does allow NOS/VE to run on larger configurations. The recommended maximum number of drivers (I/O channels) is eight for CYBER 810/830 systems.

Dual State Minimum Hardware Requirements

The following list identifies the minimum hardware required to support NOS/VE in the dual state environment.

1. 8 megabytes of memory (this assumes no activity on NOS or NOS/BE except for those functions necessary to support NOS/VE, such as interactive support, print output, etc.). This is a minimum memory requirement for any dual state configuration. The 8 megabytes should be partitioned such that NOS/VE uses 6 megabytes and NOS or NOS/BE uses 2 megabytes.
2. 15 peripheral processors as a minimum on all machines with the exception of the CYBER 810/830 which requires a 20 peripheral processor configuration.
3. Two disk controller channels; one dedicated as a minimum for NOS/VE and one dedicated to NOS or NOS/BE as a minimum. If you use a 7155 controller and an 885 disk unit, the controller must be a model B (or a model A with option 65290-1 installed. This option is indicated by tag number FV715-A.) for large sector formatting.
4. Disk storage capacity; 350 megabytes as a minimum for NOS/VE and one physical disk unit for NOS or NOS/BE.
5. At least one tape controller and tape unit is required per machine. The controller/unit can be shared between NOS or NOS/BE and NOS/VE.
6. In most cases, additional channels are required and are either added as part of the first peripheral processor increment on the CYBER 840 and larger machines, or are required before a peripheral processor upgrade can be added to the CYBER 810/830 machines.
7. In all configurations, the CC634B or CC598B console is required as the NOS/VE operator interface. A CC545 console is required for NOS/BE.
8. If files are to be printed out at the central site, the site should have a printer with ASCII capability.

This list of minimum hardware requirements lets you install and initialize NOS/VE and accommodate a minimum amount of usage. Additional memory and disk controllers/units may be necessary to support any activity on NOS or NOS/BE or increased activity on NOS/VE.

NOS/VE Initiation

In order to initiate deadstart of NOS/VE, the disk and tape channels and equipment used by NOS/VE must be made unavailable to NOS using the DOWN commands and the NVE subsystem must be initiated. These steps may occur automatically after NOS deadstart completes or manually by using operator commands.

Specific details are:

- To automatically down the channels used by NOS/VE, include IPRDECK DSD entries which will DOWN the appropriate channels. The commands will be executed once NOS deadstart completes and before subsystems are initiated.
- To automatically initiate the NVE subsystem, include an IPRDECK ENABLE entry for the NVE subsystem. Typically, control point 3 is specified on the ENABLE entry. In addition, the name of the NVE subsystem startup procedure must be NVE. This is set when executing SETVE, as documented in the NOS/VE Installation Handbook.
- To select the control point for NVE but not have it initiated automatically, use an IPRDECK DISABLE entry. This ensures that the NVE subsystem always runs at the same control point.

Refer to the NOS Version 2 Analysis Handbook for more information on the DSD, ENABLE, and DISABLE IPRDECK entries.

VEIAF Family Name Selection

Interactive users who access NOS/VE through NAM are normally assigned to the default NOS/VE login family. Sites with multiple NOS/VE families can modify the MAP_NOS_FAMILY_TO_NOSVE procedure to assign a user to a different NOS/VE family. The validated NOS user and family name is passed to the procedure, which then returns the user's assigned NOS/VE family and user name to which the user is assigned. This procedure is located in deck IIM\$NAM_PASSON on the Dual State Source Library.

NOTE

If this procedure is modified, users will be required to specify the correct NOS family and user names on the SET_LINK_ATTRIBUTES command, since the returned NOS/VE family and user names will be used for that user's link attributes. Link attributes are used for these NOS/VE commands: CREATE_INTERSTATE_CONNECTION, GET_FILE, REPLACE_FILE, and PRINT_FILE with the DUAL_STATE_ROUTING_PARAMETER (DSRP).

NOS SETVE Modification

If you change the default password of the NOS user name SYSTEMX, then you must alter the USER statement in the SETVE procedure.

To update the SETVE procedure with the correct USER statement, follow these general steps:

1. Modify the Dual State Source Library file on NOS/VE.
2. Convert the library to text.
3. Move the converted library to NOS and rebuild dual state.
4. Create a new deadstart tape.

As an alternative to the above method, you can use the following general steps to temporarily change SETVE:

1. Modify the SETVE procedure on the system.

```
RETURN, SYSTEM, SETVE.  
COMMON, SYSTEM.  
GTR, SYSTEM, SETVE. PROC/SETVE
```

Edit file SETVE to update the USER statement.

2. Put the updated SETVE procedure on a new deadstart tape.

```
SYSTEM, DST, SYSTEM, SETVE, NEW, 0, density.
```

Replace density with the density of the new deadstart tape (HY, HD, PE, or GE). A tape request is issued for an unlabeled tape with VSN=NDT. The tape may be assigned from the system console using the following command:

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

Replace est with the EST ordinal number of the tape unit where the new deadstart tape is mounted. LIBEDIT output is written to a local file named SYSLIST.

Use the new deadstart tape the next time you deadstart and SETVE will run correctly.

NOS RUNJOBS Modification

If you change the default password of the NOS user name SYSTEMX, then you must alter the USER statements in the RUNJOBS procedure.

To update the RUNJOBS procedure with the correct USER statement, follow these general steps:

1. Modify the Dual State Source Library file on NOS/VE.
2. Convert the library to text.
3. Move the converted library to NOS and rebuild dual state.
4. Create a new deadstart tape.

As an alternative to the above method, you can use the following general steps to temporarily change RUNJOBS:

1. Modify the RUNJOBS procedure in the ULIB NVELIB.

```
RETURN, SYSTEM, OLD, LGO, DSTLIB.
COMMON, SYSTEM.
GTR, SYSTEM, OLD, U.ULIB/NVELIB
GTR. PROC/RUNJOBS
```

Edit file LGO to update the USER statement.

2. Place the updated NVELIB in file DSTLIB on the user name from which you invoke NOS/VE.

```
PURGE, DSTLIB/NA.
DEFINE, DSTLIB/CT=PU.
LIBEDIT, N=DSTLIB, U, I=0.
RETURN, DSTLIB.
```

The next invocation of NVE will use the RUNJOBS procedure stored in file DSTLIB.

NOTE

When upgrading NOS/VE in a dual state environment, any local modifications to file DSTLIB should be migrated to the new NVELIB prior to deadstarting the new NOS/VE level for the first time. Then the file DSTLIB should be purged or renamed. If this is not done, it is possible to use an old or incompatible NVELIB when deadstarting your new level of NOS/VE.

NOS ACCFILE

The following template files are used by NOS/VE Interstate Communications and the NOS/VE commands PRINT_FILE, GET_FILE, REPLACE_FILE, and CREATE_INTERSTATE_CONNECTION to generate partner jobs that are submitted to NOS:

- ICACCNT Created in the ACCFILE procedure; used for interstate communications jobs and the CREATE_INTERSTATE_CONNECTION command.
- PRACCNT Created in the RUNJOBS procedure; used when the DUAL_STATE_ROUTE_PARAMETERS parameter is specified on the PRINT_FILE command.
- RHACCNT Created in the ACCFILE procedure; used for GET_FILE and REPLACE_FILE.

These template files are created during NOS/VE deadstart. The released template files have the following partner job template:

```
&JOB .
USER, &USER, &PASSWORD, &FAMILY .
CHARGE, &CHARGE, &PROJECT .
```

Parameter	Description
&JOB	<p>The GET_FILE and REPLACE_FILE commands cause the following default command to be used:</p> <pre>P, T5000 .</pre> <p>This job executes in the communication task service class.</p> <p>The CREATE_INTERSTATE_CONNECTION command uses the PARTNER_JOB_CARD parameter, if one was specified; otherwise, it uses the default job command:</p> <pre>FASLAVE .</pre> <p>This job executes in the batch service class.</p> <p>When the DUAL_STATE_ROUTE_PARAMETERS parameter is specified on the PRINT_FILE command, the following default job command (NOS/VE login user name) is used:</p> <pre>username .</pre> <p>This job executes in the communication task service class.</p> <p>Interstate communications does no substitution; a NOS job command must be supplied.</p>
&USER	<p>The user name value specified for the SET_LINK_ATTRIBUTES command is substituted. If the &ORGUSER field is used, the SET_LINK_ATTRIBUTES command is ignored. If no SET_LINK_ATTRIBUTES command is issued or the &ORGUSER field was used, then the current NOS/VE user name is used.</p>

Parameter	Description
&PASSWORD	The password value specified for the SET_LINK_ATTRIBUTES command is substituted. If no SET_LINK_ATTRIBUTES command was issued, then the current NOS/VE password is used.
&FAMILY	The family name value specified for the SET_LINK_ATTRIBUTES command is substituted. If the &ORGFAMILY field is used, the SET_LINK_ATTRIBUTES command is ignored. If no SET_LINK_ATTRIBUTES command is issued or the &ORGFAMILY field was used, then the current NOS/VE family is used. If the family name is NVE, then the NOS family under which the NVE subsystem is executing is substituted.
&CHARGE	The value specified by the CHARGE parameter of the SET_LINK_ATTRIBUTES command is used. If the &ORGCHARGE field is used, the SET_LINK_ATTRIBUTES command is ignored. If no SET_LINK_ATTRIBUTES command was issued or the &ORGCHARGE field was used, then substitution depends on the NOS/VE user's project required validation. If this validation is true, the NOS/VE user's account name is substituted. If this validation is false, then the two characters *. (asterisk period) are substituted.
&PROJECT	The value specified by the PROJECT parameter of the SET_LINK_ATTRIBUTES command is used. If the &ORGPROJECT field is used, the SET_LINK_ATTRIBUTES command is ignored. If no SET_LINK_ATTRIBUTES command was issued or the &ORGPROJECT field was used, then substitution depends on the NOS/VE user's project required validation. If this validation is true, the NOS/VE user's project name is substituted. If this validation is false, then a blank character is substituted.

NOS ACCFILE Modification

You may want to modify the partner job templates in the procedures ACCFILE and/or RUNJOBS, for example, to modify these job templates for local accounting or for taking out the &FAMILY if you are running with one family on NOS. If the &FAMILY is left in and if the family defined for your site for NOS is different from the family NOS/VE, then each user must enter a SET_LINK_ATTRIBUTES command in order to execute any dual state commands (GET_FILE, REPLACE_FILE, CREATE_INTERSTATE_CONNECTION). To change the ACCFILE or RUNJOBS procedure, follow these general steps:

1. Modify the Dual State Source Library on NOS/VE.
2. Convert the library to text.
3. Move the converted library to NOS and rebuild dual state.
4. Create a new deadstart tape.

As an alternative to the above method, you can use the following general steps to temporarily change the job templates:

1. Modify the ACCFILE and/or RUNJOBS procedure in the ULIB NVELIB.

```
RETURN, SYSTEM, OLD, LGO, DSTLIB.
COMMON, SYSTEM.
GTR, SYSTEM, OLD, U. ULIB/NVELIB
GTR. PROC/ACCFILE, RUNJOBS
```

Edit file LGO to update the job templates.

2. Place the updated NVELIB in file DSTLIB on the user name from which you invoke NOS/VE.

```
PURGE, DSTLIB/NA.
DEFINE, DSTLIB/CT=PU.
LIBEDIT, N=DSTLIB, U, I=0.
RETURN, DSTLIB.
```

The next invocation of NVE will use the new procedures stored in file DSTLIB.

NOTE

When upgrading NOS/VE in a dual state environment, any local modifications to file DSTLIB should be migrated to the new NVELIB prior to deadstarting the new NOS/VE level for the first time. Then the file DSTLIB should be purged or renamed. If this is not done, it is possible to use an old or incompatible NVELIB when deadstarting your new level of NOS/VE.

FDBF - FORTRAN Data Base Facility Version 1

This section describes the following:

- Product Dependencies
- Unique Parameters

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

Parameter	Description
------------------	--------------------

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 section describes the following:

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

FSEEX and SMFEX Implementations

There are two implementations for 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 an 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 50000₈ words, the SMFEX scheduler processes approximately 3000₈ words. SMFEX can process editing transactions most effectively if the hardware configuration provides extended memory in the amount of 3000₈ words per editing user. Extended memory can be ECS, STORNET, ESM, LCME, 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 54000₈ to 70000₈ 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 between the following choices: stand-alone FSEEX which uses a large resource per user, or combined FSEEX plus SMFEX which 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 Subsystem Initiation, at the beginning of this chapter, for information about the subsystem startup files and SMF subsystem initiation.

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

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

The sequence of DMB, EXIT, and SMFEX,RECOVER commands is mandatory for valid subsystem shutdown and abort processing.

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

```
SMFEX,n.
```

where 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 4000₈ words.

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

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

where m is the extended memory field length in units of 1000₈ 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	Description
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 _g 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.

Site-Defined Teach File

If you want to define an FSTEACH file different from the released FSTEACH file, create the file and place it on user name LIBRARY. The file should be named the model name of the terminal used at your site prefixed by the letter T. For example, for a 722 terminal, name the file T722.

FSE responds to a TEACH command by looking for a file whose name is the current terminal model name prefixed by a T (for example, T722 for a 722 terminal) on user name LIBRARY. FSE uses this file as FSTEACH. If no T file is found, FSE uses FSTEACH by default.

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.

SMFSTAT Procedure

FSE installation creates an indirect access permanent file on the installation user name that contains a procedure called SMFSTAT. This procedure provides some statistics for the Screen Management Facility (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 installation user name.
2. Enter this command:

```
BEGIN, , SMFSTAT.
```

FSE comes up, and you may then enter the GET DATA (GD) directive. If SMF 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, FTN5 (FORTRAN Extended Version 4, FORTRAN Extended Version 4 with Interactive Option, FORTRAN 5)

This section describes the following:

- MODEL Parameter
- Installation Parameters for FTN4 and FTNTS
- Installation Parameters for FTN5

MODEL Parameter

FTN4, FTNTS, and FTN5 reference the MODEL parameter (refer to the TEXT and TEXTIO section in this chapter). 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.

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 parameters 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 a listing of the parameters by assembling FTN5TXT and/or 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 FTN5TXT).

Reinstall the compiler and CCG whenever you change parameters in OPTIONS (refer to the Build Dependency Chart in chapter 6). You can revise installation parameters in COMFCIP during the installation of FTN5, if you reassemble both FTN and INIT00.

IAF - Interactive Facility Version 1

This section 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	Description
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 Subsystem Initiation, at the beginning of this chapter, for information about subsystem startup files and subsystem initiation. IAF initiation consists of these three procedures: IAF, IAFTM, and IAFTR.

When IAFTM or IAFTR is used, it stores a copy of TRACIAF under SYSTEMX (user index 377777₈) for subsequent use. For additional information regarding the trace utility, refer to the NOS Version 2 Analysis Handbook.

NOTE

The composite OPL deckname for the IAF procedure is IAFP.

Special Notes

- RDFEX and IAF share the decks IAFEX and 1TM in the composite OPL (OPLpsrout). Both products must be rebuilt if user code is applied to either of these decks, and the user code must be included with both the RDFEX and IAF build procedures.
- Procedure file TRACIAF, which uses a NETOPS USER command and the default CHARGE command, contains commands to process the trace file. Modify TRACIAF only if you need to change the USER and CHARGE command parameters. Modify decks IAFTM and IAFTR and place the modifications on a USER file.
- 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 command. 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 the NOS Version 2 Analysis Handbook for a description of the IAFEX command.

ITF - Interactive Transfer Facility Version 1

This section describes the following:

- Product Dependencies
- ITF Command

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 on the installation user name. After all products that modify this file are installed, you can use the SYSGEN(MOVE) command to move the startup master file to the proper user name (refer to the NAM5 section in this chapter). 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 a USER file for the ITF installation procedure.

Before using ITF, you must ensure that the LCF section of 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):

```
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 Configuration Information in the RHF section of this chapter):

```
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 Description

Parameter	Description
CO=co	The maximum number of terminals connected per remote host. This value can range from 1 through 128. Default is 64.
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.
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.
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. This value can range from 1 through 128. Default is 128.

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:

$$0 < PI \leq MAXACN$$

$$0 < CO \leq MAXTCN$$

$$0 < TE \leq \min(PI, MAXACN) * \min(CO, MAXTCN)$$

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:

$$(2 * MAXACN) + (MAXACN * MAXTCN) \text{ words}$$

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.

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 section 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 a USER file in the installation procedure for LCS3.

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

Table sizes and central memory requirements are shown in table 7-7.

Table 7-7. 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 a USER file when the LCS3 installation procedure is run.

```
*DEFINE CELCOPY
```

The central memory requirements for this version of the LCS are increased by approximately 20400₈ 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 a USER file when the LCS3 installation procedure is run.

```
*DEFINE COLUMN6
```

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

LOADER - CYBER LOADER Version 1

This section describes the following:

- Unique Parameters
- Installation Parameters

Unique Parameters

Parameter	Description
-----------	-------------

PRESET The PRESET parameter sets the default central memory presetting options. The default presetting for the binaries you receive depends on the information you provided in the OIP. The default for this parameter is ZERO. If the PRESET parameter is specified on the LDSET command, it will override this default setting. The value NONE requires no presetting; same as ZERO for CM.

Value	Preset (Octal)				
NONE	-	-	-	-	-
ZERO	0000	0000	0000	0000	0000
ONES	7777	7777	7777	7777	7777
INDEF	1777	0000	0000	0000	0000
INF	3777	0000	0000	0000	0000
NGINDEF	6000	0000	0000	0000	0000
NGINF	4000	0000	0000	0000	0000
ALTZERO	2525	2525	2525	2525	2525
ALTONES	5252	5252	5252	5252	5252
DEBUG	6000	0000	0004	0040	0000

MAP This parameter sets the default LOADER MAP option. The default is OFF. If the MAP parameter is specified on the LDSET command or the MAP command is used, it will override this default setting.

Value	Description
OFF	No map
PART	Statistics, block map
ON	Statistics, block map, entry point cross-reference
FULL	Statistics, block map, entry point map, entry point cross-reference

Parameter	Description						
FLMSG	This parameter determines if LOADER issues a dayfile message indicating the field length required for loading and execution. The default is YES.						
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>Dayfile message is not issued</td> </tr> <tr> <td>YES</td> <td>Dayfile message is issued</td> </tr> </tbody> </table>	Value	Description	NO	Dayfile message is not issued	YES	Dayfile message is issued
Value	Description						
NO	Dayfile message is not issued						
YES	Dayfile message is issued						

NOTE

When building LOADER, 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 Update.

Parameter	Default	Description								
IP.FLINC	4000B	Amount of field length increase if additional field length is required for table construction by LOADER. Acceptable values are multiples of 100g.								
IP.LDBG	0	If nonzero, conditional code to aid in debugging the loader is assembled.								
IP.LDER	1	Error processing by the loader; one of the following values. <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Abort on all errors (ERR=ALL).</td> </tr> <tr> <td>1</td> <td>Abort on fatal errors (ERR=FATAL).</td> </tr> <tr> <td>2</td> <td>No abort if abort is possible (ERR=NONE).</td> </tr> </tbody> </table>	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).
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).									
IP.LRT	0	If nonzero, a message giving various time and memory measurements is issued to the dayfile.								
IP.REW	1	Specifies whether the file is rewound prior to beginning of load; one of the following values. <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>File is not rewound.</td> </tr> <tr> <td>1</td> <td>File is rewound.</td> </tr> </tbody> </table>	Value	Description	0	File is not rewound.	1	File is rewound.		
Value	Description									
0	File is not rewound.									
1	File is rewound.									

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

MAP Subsystem

This section describes the following:

- MAP Installation Procedures
- MSSSI Validation Requirements
- Procedures for Initiating MSSSI

MAP Installation Procedures

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

The MAP subsystem procedures and their build parameters are as follows:

Procedure	Build Parameter
AP1I – Online diagnostics	MEMSIZE
AP1L – Online diagnostics	MAPTYP, ADD, MUL, DIV, SQRT
MMCL – Macro control library	None
MSSSI – MAP subsystem	BLDMLIB, MSAMLIB

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

NOTE

The MSSSI procedure requires 20K of ECS/UEM and the AP1I procedure requires 70K of ECS/UEM.

MSSSI Validation Requirements

The SYSGEN installation procedure for installing MSSSI requires 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.

Procedures for Initiating MSSI

Control Data provides three procedure files for initiating MSSI:

MAPCH for initiating MAP IV-20/21

MAPCMI for initiating MAP IV-23/25

MAPECS for initiating MAP III

MCS - Message Control System Version 1

This section describes the following:

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

Unique Parameters

Parameter Description

TRACE	To activate the MCS debug facility, specify the keyword TRACE on the call to the MCS installation procedure.
-------	--

Installation Parameters

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

Parameter	Default	Description
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	Checkpoint.
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 Subsystem Initiation, at the beginning of this chapter, for information about the MCS subsystem startup procedure and 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 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 in a USER file for the MCS installation procedure.
- Users must be validated to access MCS (refer to MODVAL in the NOS Version 2 Administration Handbook).

MSE - Mass Storage Extended Subsystem

This section describes the following installation options for MSE:

- Unique Parameters
- Configuration Information
- Common Deck Parameters

Unique Parameters

Parameter	Description
SAVELIB	To save M86LIB as a direct access file, specify the keyword SAVELIB on the call to the MSE installation procedure.

Configuration Information

To configure the Mass Storage Extended (MSE) subsystem, create the following:

- **MSECONF** file. This file defines the components of your 7990 configuration. Some of the statements in the file reference your EQPDECK entries for the 7990. The MSECONF file is stored as a direct access file on user name SUBFAM0 (user index 377760B). A sample MSECONF file is provided with the release materials. This file should be edited on the installation user name then moved to user name SUBFAM0.

For a first-time installation, ATTACH the example file from user name SUBFAM0 directly. If you are installing MSE for the first time during an upgrade or component order installation, execute the SYSGEN(MSE1) procedure from the installation user name to load a copy of the MSECONF, MSE, and MSES LAV files to the installation user name.

To move the MSECONF file to user name SUBFAM0, execute the command SYSGEN(MOVE,MSECONF,SUBFAM0,,Y,PERMIT) from the system console when directed to in chapter 2, 3, or 4.

- **SS EQPDECK** entry. This entry defines the connection of the 7990 controller to your mainframe.
- **MSE startup procedures.** These procedures define how the MSE subsystem should be initiated. Two procedures are released: one for executing MSE in a single mainframe environment (file name MSE) and one for executing MSE in slave mode in a multmainframe environment (file name MSES LAV). The MSE startup procedures are indirect access files stored on user name SYSTEMX (user index 377777B). These files should be edited on the installation user name then moved to user name SYSTEMX.

For a first-time installation, GET the example files from user name SYSTEMX directly. If you are installing MSE for the first time during an upgrade or component order installation, execute the SYSGEN(MSE1) procedure from the installation user name to load a copy of the MSE, MSES LAV, and MSECONF files to the installation user name.

To move either file to user name SYSTEMX, execute the command SYSGEN(MOVE,filename,SYSTEMX,,Y,PERMIT) from the system console when directed to in chapter 2, 3, or 4.

The file PFGMSE1 on the permanent file tapes contains the MSECONF, MSE, and MSESLAV files. For more information about the MSE configuration file, SS EQPDECK entry and MSE startup procedures, refer to the Mass Storage Extended Subsystem and Deadstart Decks sections in the NOS Version 2 Analysis Handbook.

Common Deck Parameters

COMBFAS Parameters

COMBFAS contains the following parameters used by the MSE executive (SSEXEC). Assemble CALLFAS to obtain a listing of COMBFAS.

Parameter	Default	Description
MAXCHERR	2	Number of channel errors allowed per hour.
MAXCTUNIT	1	Number of controllers allowed in the Unit Device Table. Maximum is 8.
MAXSMM1	1	Must equal MAXSMUNIT - 1.
MAXSMUNIT	2	Number of storage modules allowed in the Unit Device Table. Maximum is 8.

COMXIPR Parameters

COMXIPR contains the following parameters used by the MSE executive (SSEXEC). Assemble CALLFAS to obtain a listing of COMXIPR.

Parameter	Default	Description
NUMRB	9	The number of file staging request blocks available to a slave mainframe.
NUMSLV	3	The number of slave mainframes for which master SSEXEC can service file staging requests.
SLAV\$INTV	60	The number of seconds SSEXEC waits with no signal from a slave mainframe before assuming that the slave executive has terminated.
SLRP\$INTV	5	The number of seconds SSEXEC waits to look for new staging requests from slave mainframes.

NAM5/NAM5D - Network Access Method Version 1

This section describes the following installation options for NAM:

- Configuration Information
- Site-Defined User Names
- Special Notes
- Unique Parameters
- NAM Procedure File
- USER File Directives
- Network Startup Master File

Configuration Information

To configure the Network Access Method, create the following:

- **NDL file.** Entries in a Network Definition Language (NDL) file are made in two sections, the Network Configuration File (NCF) portion and the Local Configuration File (LCF) portion. The Network Definition Language Processor (NDLP) compiles the NCF portion of the NDL into the NCFFILE and the LCF portion into the LCFFILE. NCFFILE and LCFFILE reside as direct access permanent files on the network administration user name. These files should be private files with read permission given to the network operations user name. NCFFILE and LCFFILE can also be public or semiprivate.

If you are installing a NAM/CCP network, the NDL file must contain both an NCF portion and an LCF portion. If you are installing a NAM/CDCNET network, only the LCF portion is required. More details on the two portions of the NDL are given below.

- **Network Configuration File (NCFFILE).** The NCF portion of an NDL file describes the connections made between 255x NPUs (both the trunks and X.25 direct lines) and the connections between the lines and terminals and each NPU. It also defines which hosts can load which NPUs and the paths available for application-to-application connections between hosts. Entries are made to define a NAM/CCP network, the Printer Support Utility (PSU), and PTF/QTF host-to-host logical links or X.25 lines for 255x networks. (PSU entries are needed only if the printer is connected to a 255x NPU.)

When creating the NCF, be sure the values for coupler node numbers in the NPU definitions match the 255x ND parameter on the EQPDECK entries for each 255x and that each NPU variant referenced in the NCF exists in the CCP network load file (NLFFILE).

- **Local Configuration File (LCFFILE).** The LCF portion of the NDL file defines the network applications that execute on the mainframe as well as the attributes of application-to-application connections between hosts. It also defines login and connection attributes for lines and terminals connected to a 255x NPU. Entries are made to define a NAM/CCP network, a NAM/CDCNET network, the Printer Support Utility (PSU), and PTF/QTF INCALL, OUTCALL, and APPL definitions for CDCNET and 255x networks. (PSU entries are needed only if the printer is connected to a 255x NPU.)

The release materials contain a sample NDL file, NDLDATA, which is installed to the network administration user name. This file defines a simple one-NPU network that includes the definitions for the PSU printer and all CDCNET applications. This file may be examined by logging in to the network administration user name or by executing SYSGEN(NDL1) from the installation user name to load NDLDATA (and the compiled version of NDLDATA, NCFFILE and LCFFILE) to disk.

To compile an NDL, execute the following steps:

1. Log in to the network administration user name and update NDLDATA using an available editor. Leave NDLDATA local and rewound.
2. If you are updating the NDL as part of a system upgrade (and have therefore not deadstarted the system containing the new level of NDLP), access the new version from GLOBLIB:

```
ATTACH,GLOBLIB/UN=insun.
LIBRARY,GLOBLIB/A.
```

3. Make permanent files to receive compiled NCFFILE and LCFFILE.

```
PURGE, NCFTEMP, LCFTEMP/NA.
RETURN, NCFFILE, LCFFILE.
DEFINE, NCFFILE=NCFTEMP, LCFFILE=LCFTEMP.
PERMIT, NCFTEMP, netopun=R.
PERMIT, LCFTEMP, netopun=R.
```

4. Execute NDLP:

```
NDLP, I=NDLDATA, L=listing.
```

Replace listing with the name of the file to receive the NDLP output.

```
RETURN, NCFFILE, LCFFILE.
```

5. Change the temporary names for the NCFTEMP and LCFTEMP files in Step 6 of chapter 3 or 4, whichever is appropriate. Use the commands below:

```
PURGE, NCFOLD, LCFOLD/NA.
CHANGE, NCFOLD=NCFFILE, NCFFILE=NCFTEMP.
CHANGE, LCFOLD=LCFFILE, LCFFILE=LCFTEMP.
```

- EQ EQPDECK entry for CCP. This entry defines the connection of each 255x NPU to your mainframe. The ND parameter is referenced by the NODE parameter of the NDL COUPLER statement.
- EQ EQPDECK entry for CDCNET. This entry defines the connection of each Mainframe Device Interface or Mainframe Terminal Interface to your mainframe. The ND and NT parameters are (optionally) referenced in CDCNET DI system configuration files.

Refer to the Network Definition Language (NDL) Reference Manual for more information on the NDL and NDLP. Refer to the Deadstart Decks section of the NOS Version 2 Analysis Handbook for more information about the EQ EQPDECK entries. Refer to the CDCNET Configuration and Site Administration Guide for more information about CDCNET configuration files.

Site-Defined User Names

If you are changing the default Control Data installation user names (in particular, NETOPS or NETADMN), note the following:

- If the network administration user name is changed, file NAMSTRT must be altered. NAMSTRT contains a parameter called NETUN2 that determines where the system expects such files as NCFFILE, LCFFILE, NLFFILE, and TCPHOST. This example shows the line to change:

```
PARAM (NETUN2=NETADMN)          CONFIGURATION/LOAD FILE USER NAME.
```

The NETUN2 parameter exists in the six basic NAMSTRT records (INIT-MRECOV). You must change it in each record that you use.

- If the network operations user name is changed, you must tell NAM what the new user name is the next time you initiate NAM (in Step 7:

Deadstart the New System). This can be accomplished by two methods:

1. If you moved NAMSTRT to the new network operations user name, NAM should ask for a CFO command when it is initiated. Enter the following commands from the system console:

```
CFO, NAM.UN=netopun, PW=password.
CFO, NAM.GO.
```

These commands tell NAM the new user name on which NAMSTRT resides.

2. If you did not move NAMSTRT to the new network operations user name, you must execute the NAMNOGO procedure to initiate NAM. This gives you the opportunity to tell NAM the new user name and password. Enter the following commands from the system console:

```
CFO, NAM.UN=netopun, PW=password.
CFO, NAM.GO.
```

These commands tell NAM the new user name on which NAMSTRT resides.

In either case, NAM updates its files so that the next time it is brought up, it automatically finds NAMSTRT on the new network operations user name.

Special Notes

- The NAM installation procedure installs the following NAM components and utilities:
 - AIP (Application Interface Program)
 - COLLECT (Collect Network Dumps)
 - CS (Communications Supervisor)
 - DLFP (Debug Log File Processor)
 - LFG (CCP Load File Generator)
 - LISTPPM (Format PIP Memory Dumps)
 - NAMI (Network Initialization Program)
 - NDA (NPU Dump Analyzer)
 - NDLP (Network Definition Language Processor)
 - NIP (Network Interface Program)
 - NS (Network Supervisor)
 - NVF (Network Validation Facility)
 - PIP (Peripheral Interface Program)
 - QTRM (Queued Terminal Record Manager)
 - TVF (Terminal Verification Facility)
- NAM5 binaries are released on the deadstart tape in non-debug/trace mode. Control Data provides debug/trace mode binaries of NAM5 on the permanent file tapes. To obtain the debug/trace NAM5 binaries, use the SYSGEN(SWAP) function described in chapter 8.
- The installation procedure retrieves the startup 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. Executing the SYSGEN (MOVE) utility automatically moves the files to SYSTEMX and the network operations user name. Other optional product installation procedures modify NAMSTRT (PSU, RBF, PTF/QTF with NAM interface, CDCNET Host Applications, TCP/IP/FTP/TELNET and ITF). Thus, you should not move NAMSTRT until these optional products have been installed.
- The NAM1 Host Application Programming Reference Manual describes AIP, CS, DLFP, NIP, NS, PIP, and QTRM. The NAM/CCP Terminal Interfaces Reference Manual describes TVF. The Network Definition Language Reference Manual describes NDLP. The NOS Version 2 Analysis Handbook describes COLLECT, LFG, LISTPPM, NAMI, NDA, and NVF.
- There are execution time interdependencies between NAM and CCP. These interdependencies are established in file USERBPS (refer to CCP/CROSS in this chapter).
- 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 JOB_CS, JOB_NIP, JOB_NS, JOB_NVF, and JOB_TVF 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 Host Application Programming Reference Manual.

Unique Parameters

Parameter	Description
NOTRACE	To disable trace/log file creation for CS, NS, NVF, and TVF, specify the keyword NOTRACE on the call to the NAM5 procedure. (This option is not available for NAM5D.)

NAM Procedure File

Refer to Subsystem Initiation, at the beginning of this chapter, 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 Version 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, MINIT, MRECOV, MULTI, RECOVR, and RESTRT) and seven job records (JOB COL, JOB CS, JOB NVF, JOB TVF, JOB PUR, and JOB NIP). 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 COLLECT, CS, NS, NVF, and TVF. NIP is started at the NAMI control point.

Based on JOB and PARAM statements in the INIT record, other network applications (CDCNET Host Applications, TCP/IP/FTP/TELNET, ITF, PSU, PTF/QTF with NAM interface, and RBF) are also routed to the input queue to begin execution.

USER File Directives

To assemble the following features into NAM, include directives of the form

```
*DEFINE name
```

in a USER file for the NAM5 installation procedure, where name is one of the following values:

Name	Significance when Defined
DEBUG	Code to aid in debugging and maintenance in NIP and PIP is generated. The following list shows the effect of DEBUG on NAM components. <ul style="list-style-type: none"> • CS, NIP, 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 AIP and NIP. 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 NIP and PIP in local file ZZZZZDN.

NOTE

DEBUG and STAT are defined by the release defaults. To remove the definitions, specify NOTRACE on the call to the NAM5 installation procedure.

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 (NAMSTRT) is in standard NOS text format and is automatically installed on the installation user name by SYSGEN. You can modify the file using a text editor or through Update directives against the NAM program library.

A comment can follow the parameters on all statements and directives.

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	Description
type	Type of record; use PARAM for parameter records and JOB for job skeleton records.
title	Text string of 1 through 50 characters used 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) terminated by a zero byte. Refer to the NOS Version 2 Analysis Handbook for a description of parameter records available with the released system.

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(keyword1=value1, . . . , keywordn=valuen)
```

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 NCFN in any job skeleton record is replaced by the string NCFFILE.

```
PARAM(NCFN=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, at1, at2, at3])
```

Parameter	Description								
at _i	Job attribute. Any of the following are valid: <table border="1" data-bbox="500 762 1372 1050"> <thead> <tr> <th>at_i</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>DI</td> <td>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.</td> </tr> <tr> <td>NS</td> <td>If specified, the job record is routed with the Network Supervisor service class (SCL=NS).</td> </tr> <tr> <td>SY</td> <td>If specified, the job record is routed with system origin privileges.</td> </tr> </tbody> </table>	at_i	Description	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.	NS	If specified, the job record is routed with the Network Supervisor service class (SCL=NS).	SY	If specified, the job record is routed with system origin privileges.
at_i	Description								
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.								
NS	If specified, the job record is routed with the Network Supervisor service class (SCL=NS).								
SY	If specified, the job record is routed with system origin privileges.								
name	Name of job skeleton record.								
ssname	Subsystem name if program is a subsystem (not required for NIP).								
type	First two or more characters of an application or program name, such as CS, NIP, NS, and so on. (Only the first two characters are used.)								

The rules for format of a PARAM directive apply to the JOB directive.

Example:

```
JOB(JOBNIP,NIP) NAM (NIP) JOB.
```

Refer to figure 7-8 for an example of a parameter record that contains both PARAM directives and JOB directives.

```

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(NETUN2=NETADMIN) CONFIGURATION/LOAD FILE USER NAME.
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(ZZINACT=10) MINS FOR TERM INACT SUPV MESSAGE.
PARAM(ZZMC=500) MESSAGE COUNT BEFORE RELEASE TRACE FILE.
PARAM(ZZRUNCT=3) MAX NBR OF PGM RUNS WITH NO OPER ACTION.
PARAM(ONETAPE=YES) COLLECT HOST AND NPU DUMPS TO ONE TAPE.
.*
.*
JOB(JOBPUR,CO,SY,NS) COLLECTOR JOB.
JOB(JOBNIP,NIP) NAM (NIP) JOB.
JOB(JOBNVF,NV,SY,NS) NVF JOB.
JOB(JOBNS,NS,SY,NS) NS JOB.
JOB(JOBCS,CS,SY,NS) CS JOB.
JOB(JOBTVF,TV,SY,NS) TVF JOB.
.*

```

Figure 7-8. Example of Parameter Record

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.
MFN	Master file name.
OIN	Old network invocation number.
PWM	Password for master file user name.
UNM	Master file user 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.
xxS0FIL	ZZZZZSN.
xxT0FIL	ZZZZZDN.

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.

Refer to figure 7-9 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,ETC).
.*      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)
.*
.* PURGE OLD LEVEL-2 TRACE FILES.
.*
PURGE (ZZNI OIN, ZZNI CIN/NA)

```

Figure 7-9. Example of Job Skeleton Record (Continued)


```

.*
.*  START NIP.
.*
RFL(60000)
NIP(NIN=CIN, ISTEP=NIISTP, FSTEP=NIFSTEP, MC=ZZMC, INACT=ZZINACT)
.*
.*  NIP NORMAL TERMINATION - CHECK FOR ABNORMAL CONDITIONS.
.*
RFL(0)
SKIP(BAILOUT)
.*
EXIT.  NIP
.*
.*
.*  NIP FAILED - SAVE RUN DATA IF REQUIRED.
.*
.*
.*
DISPLAY(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(NIT0FIL, ZZNI CIN/NA)
RETURN(OUTPUT)
WRITER(OUTPUT)
EXIT.  NIP
.*
.*  SAVE RUN DATA IF AVAILABLE.
.*
ENDIF(SAVFILS)
.*
IF(FILE(OUTPUT, AS), NOUTPUT)
ATTACH(NIPOUT=NID0FIL/NA, M=W)
IF(.NOT.FILE(NIPOUT, AS))      DEFINE(NIPOUT=NID0FIL)
REWIND(OUTPUT)
SKIPEI(NIPOUT)
COPYEI(OUTPUT, NIPOUT)
RETURN(OUTPUT, NIPOUT)
ENDIF(NOUTPUT)
.*

```

Figure 7-9. Example of Job Skeleton Record (Continued)

```

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 (ZZZZZDMB, AS) , NOZZZDMB)
ATTACH (NIPDMP=NID2FIL/NA, M=W)
IF ( .NOT. FILE (NIPDMP, AS) )      DEFINE (NIPDMP=NID2FIL)
REWIND (ZZZZZDMB)
SKIPEI (NIPDMP)
COPYEI (ZZZZZDMB, NIPDMP)
RETURN (ZZZZZDMB, NIPDMP)
ENDIF (NOZZZDMB)
.*
IF (FILE (ZZZZZTMP, AS) , NOZZZTMP)
ATTACH (NIPZTMP=NID3FIL/NA, M=W)
IF ( .NOT. FILE (NIPZTMP, AS) )      DEFINE (NIPZTMP=NID3FIL)
REWIND (ZZZZZTMP)
SKIPEI (NIPZTMP)
COPYEI (ZZZZZTMP, NIPZTMP)
RETURN (ZZZZZTMP, NIPZTMP)
ENDIF (NOZZZTMP)
.*
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=NIT0FIL/NA, M=W)
IF ( .NOT. FILE (TRCLEV3, AS) )      DEFINE (TRCLEV3=NIT0FIL)
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)

```

Figure 7-9. Example of Job Skeleton Record (Continued)

```

RETURN (TRCLEV1, TRCLEV2, TRCLEV3)
ENDIF (NOTRACE)
.*
ATTACH (NIPLST=NIL0FIL/NA, M=W)
IF (.NOT.FILE (NIPLST, AS))      DEFINE (NIPLST=NIL0FIL)
SKIPEI (NIPLST)
NOTE (DFL, NR) /NIDA_CIN
DAYFILE (DFL)
PACK (DFL)
COPYEI (DFL, NIPLST)
.*
RETURN (OUTPUT)
WRITER (OUTPUT)
EXIT.  NIP
.EOR
.*
.*
.*  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)

```

Figure 7-9. Example of Job Skeleton Record (Continued)

```

COPYBF (INPUT, TRCLEV2)
BKSP (TRCLEV2)
SKIPR (TRCLEV2)
IF (.NOT. FILE (TRCLEV2, EOF))      WRITEF (TRCLEV2)
SETJOB (DC=NO)
EXIT.  NIPA
.EOR
.*
.*
.*  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 *NIT0FIL* 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=NIT0FIL/NA, M=W)
IF (.NOT. FILE (TRCLEV3, AS))      DEFINE (TRCLEV3=NIT0FIL)
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 7-9. Example of Job Skeleton Record

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(keyword1=value1, . . . , keywordn=valuen)
```

Parameter	Description
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
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.
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.
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.

CS may not be required in every host of a multihost network. If you do not want CS, remove the CS job statements from the network startup master file.

CS requires the following files:

Name	Description
NCF	Network configuration file created by NDLP. The permanent file name is NCFFILE. NCFFILE must be resident under the user name specified by the NETUN2 parameter. The released default for NETUN2 is NETADMN.
NRF1	Job record to be copied to the ZZZZZDN file by NETREL. This job record 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. This job record 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,INACT=ito,N1=n1,N2=n2,N3=n3,MAXFL=maxfl)

NOTE

If a value less than the minimum is supplied, the minimum value is used.

Parameter	Description						
FSTP=fstp	Option to stop all local NPUs upon network host termination. The default is YES.						
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>Do not stop local NPUs.</td> </tr> <tr> <td>YES</td> <td>Stop all local NPUs.</td> </tr> </tbody> </table>	Value	Description	NO	Do not stop local NPUs.	YES	Stop all local NPUs.
Value	Description						
NO	Do not stop local NPUs.						
YES	Stop all 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. The range is 0 to 99.						
ISTP=istp	Option to stop all local NPUs during network host initialization. The default is NO.						
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>Do not stop local NPUs.</td> </tr> <tr> <td>YES</td> <td>Stop all local NPUs.</td> </tr> </tbody> </table>	Value	Description	NO	Do not stop local NPUs.	YES	Stop all local NPUs.
Value	Description						
NO	Do not stop local NPUs.						
YES	Stop all local NPUs.						
MAXFL=maxfl	Maximum field length that NIP can reach. The range is from 61440 to 122880. The default is 98304.						
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.						
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAMI when the network operation is started. This parameter is required.						
N1=n1	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 30. The minimum value is 2.						

Parameter	Description
N2=n2	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 30. The minimum value is 4.
N3=n3	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 30. The minimum value is 2.

Use the following formula to determine a value for MAXFL for a particular configuration. (Round MAXFL to the nearest multiple of 64.)

$$\text{MAXFL} = 11096 + 700h + 6560a + 30(c+d) + 30e + m + 20(k_1w_1 + \dots + k_nw_n) + 22y + 13z + 78b_1 + 142b_2 + 206b_3$$

Variable	Description
a	Maximum number of applications with up to eight application-to-application connections.
b ₁	Maximum number of 64-word PRU buffers (N1 times the number of drivers) that can be allocated to NAM.
b ₂	Maximum number of 128-word PRU buffers (N2 times the number of drivers) that can be allocated to NAM.
b ₃	Maximum number of 192-word PRU buffers (N3 times the number of drivers) that can be allocated to NAM.
c	Maximum number of hosts in network.
d	Maximum number of NPU nodes in network.
e	Maximum number of applications to be connected at any one point in time.
h	Maximum number of host nodes.

Variable Description

$k_i w_i$ 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 downline block limit (DBL). Use one of the following values:

Value 1

$ABL \leq 2$ and $DBL < 2$.

Value ABL - 1

$ABL > 2$ and $DBL \leq 2$.

Value DBL - 1

$ABL \leq 2$ and $DBL > 2$.

Value ABL + DBL - 2

$ABL > 2$ and $DBL > 2$.

w_i Number of interactive terminals with the specified block limit.

m Maximum node number.

y Number of connected batch terminal devices.

z Number of connected interactive devices.

PRU buffers are dynamically allocated as necessary. The maximum number of buffers specified should be enough to handle the heaviest load. Specifying a large value will have no affect on network performance or resources unless there is heavy PRU traffic. 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 (N1=2), two 128-word PRU buffers (N2=2), and two 192-word PRU buffers (N3=2) if PIP supports the PRU data transfers on all three PRU buffer sizes.

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								
FDP=fdp	Forced dump option. The default is NO.								
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>NPUs are not dumped in the first 10 minutes of program execution except for the initial load.</td> </tr> <tr> <td>YES</td> <td>After the first initial load and in the absence of other NPU dumping conditions, NPUs are dumped in the first 10 minutes of program execution before loading takes place.</td> </tr> </tbody> </table>	Value	Description	NO	NPUs are not dumped in the first 10 minutes of program execution except for the initial load.	YES	After the first initial load and in the absence of other NPU dumping conditions, NPUs are dumped in the first 10 minutes of program execution before loading takes place.		
Value	Description								
NO	NPUs are not dumped in the first 10 minutes of program execution except for the initial load.								
YES	After the first initial load and in the absence of other NPU dumping conditions, NPUs are dumped in the first 10 minutes of program execution before loading takes place.								
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.								
	<table border="1"> <thead> <tr> <th>Option</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>P</td> <td>Private file.</td> </tr> <tr> <td>PU</td> <td>Public file.</td> </tr> <tr> <td>S</td> <td>Semiprivate file.</td> </tr> </tbody> </table>	Option	Description	P	Private file.	PU	Public file.	S	Semiprivate file.
Option	Description								
P	Private file.								
PU	Public file.								
S	Semiprivate file.								
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned to NAMI when the network operation is started. This parameter is required.								
RT=rt	Release trace file option. The default is YES.								
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>Trace files are not requested to be released.</td> </tr> <tr> <td>YES</td> <td>NAM requests NS programs CS and NVF to release their trace files whenever NS dumps an NPU.</td> </tr> </tbody> </table>	Value	Description	NO	Trace files are not requested to be released.	YES	NAM requests NS programs CS and NVF to release their trace files whenever NS dumps an NPU.		
Value	Description								
NO	Trace files are not requested to be released.								
YES	NAM requests NS programs CS and NVF to release their trace files whenever NS dumps an NPU.								

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 Version 2 Analysis Handbook for a description of LFG). The permanent file name is NLFFILE. NLFFILE must be resident under the user name specified by the NETUN2 parameter. The released default for NETUN2 is NETADMN.
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. The permanent file name is LCFFILE. LCFFILE must be resident under the user name specified by the NETUN2 parameter. The released default for NETUN2 is NETADMN.
NRF1 and NRF2	Described under CS, earlier in this section.

NIP5870 - 5870 Printer

This section describes the installation procedure messages for the 5870 printer.

Installation Procedure Messages

The DECKOPL installation procedure for NIP5870 contains a loader error. The following messages appear in the load map:

```
***** ERROR SUMMARY
      NE4103///DUPLICATE PROGRAM NAME FROM FILE
      PROGRAM SKIPPED --- HSTCOPY
      LAST FILE ACCESSED- HSTBIN
```

The following message appears in the dayfile:

```
NON-FATAL LOADER ERRORS - SEE MAP
```

This condition is non-fatal and does not affect the generated binaries. The frequency of occurrence is relative to this product as released. Any local code may change the frequency.

NOS and NOS2B - Network Operating System

The NOS procedure assembles all COMPASS routines; the NOS2B procedure assembles the FORTRAN and SYMPL routines.

This section describes the following:

- Configuration Information
- CALLxxx Decks
- Installation Parameters
- 819 PPU Driver Installation

Configuration Information

NOS is configured by making entries into several text records, known as deadstart decks, which reside on the deadstart tape. These decks are described below:

- **CMRDECK** (Central Memory Resident Deck). The entries in this deck describe information kept in central memory during system operation. For example, entries determine how NOS and NOS/VE will share physical mainframe memory, how much table space should be set aside for executing jobs and for input and output queue files, and the name of the system to be displayed to users on their login banner. Entries in the CMRDECK determine which EQPDECK, IPRDECK, and LIBDECK will be used. CMRDECK entries must be made in order to install the system.
- **EQPDECK** (Equipment Deck). The entries in this deck describe the hardware peripherals connected to your mainframe. Entries specify how disks, tapes, printers, network hardware, and other equipment are connected to the computer. Additional entries are used to specify how the system is to use the hardware. For example, you specify which disks will hold the NOS system file, user permanent files, temporary files, etc. Additional entries determine if a portion of the physical memory is to be used for Unified Extended Memory (UEM) and how UEM is to be allocated. For other peripherals, such as network hardware, software node numbers must be assigned which relate a physical piece of hardware to an entry in a software configuration file. EQPDECK entries must be made to install the system.
- **APRDECK** (Auxiliary Mass Storage Parameter Deck). The entries in this deck identify locations on mass storage (disks) which are unusable or flawed. Normally, no entries are required in this deck.
- **IPRDECK** (Installation Parameter Deck). The entries in this deck specify the default mode of system operation. For example, entries determine what the relative priorities of the various job classes on the system are, what the default tape density is, and what subsystems should be initiated automatically when the system is deadstarted. Normally, no IPRDECK entries are needed unless subsystem initiation must be altered.
- **LIBDECK** (Library Edit Deck). The entries in this deck specify the attributes of the programs on the deadstart tape. Primarily, the entries specify where the programs should be located; on disk, in central memory, or in UEM. Normally, this deck does not need to be altered.

CMRDECKs, EQPDECKs, APRDECKs, IPRDECKs and LIBDECKs are described in the Deadstart Decks section of the NOS Version 2 Analysis Handbook.

The deadstart tape can contain many different combinations of deadstart decks which makes it possible to have several different configurations for the same mainframe and to configure multiple mainframes on one tape. The released deadstart tape contains deadstart decks for installing on many different mainframes and hardware configurations. Tables 2-3, 2-4, and 2-5 contain an index of the decks.

CALLxxx Decks

These decks are tools for conveniently generating listings of various system common decks, for example:

CALLCPU assembles all common decks named COMCxxx.
 CALLDIS assembles all common decks named COMDxxx.
 CALLINT assembles all common decks named COMIxxx.
 CALLPPU assembles all common decks named COMPxxx.
 CALLSYS assembles all common decks named COMSxxx.
 CALLTAB assembles all common decks named COMTxxx.

Since these decks may assemble many common decks that were never intended to be assembled together (at least in a real program), assembly errors may result. These errors are normal and have always existed in these decks. The number of errors may change from release to release. The following is an example of a COMPASS command to assemble all six decks:

```
MODIFY, A, P=opl, LO=E, Z./ *EDIT CALLCPU.CALLINT
COMPASS, I, S=NOSTEXT, S=CETEXT, O=errlist, L=list.
```

Replace opl with the name of the composite OPL, replace errlist with the name of the file to contain the assembly errors (which may be ignored), and replace list with the name of the file that will contain the desired common decks.

Installation Parameters

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 the line numbers of the installation parameters you need to change.
- Put the NOS installation procedure parameter changes on a USER file 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 7-8 for brief descriptions of the NOS common decks that contain installation parameters.

Table 7-8. NOS Common Decks

Common Deck Name	A Deck That Calls the Common Deck	Description
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.
COMSMMF	CALLSYS	Multimainframe parameters.
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.
COMSVED	CALLSYS	Virtual environment definitions.
COMTNAP	CALLTAB	Valid NAM application parameters.
PPCOM	PPTXT	Maximum number of local files, number of words in a block of ECS, maximum number of EST entries, length of L-display input and output buffers, and number of mass storage devices.

COMSACC Parameters

COMSACC contains a general description of the user validation file. Assemble CALLSYS to obtain a listing of COMSACC.

Parameter	Default	Description
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 Version 2 Administration Handbook for further information on VALIDUS.
AUFN	VALINDS	Micro definition that specifies the name of the available user indexes file. User names whose user indexes are greater than AUIMX (377700 ₈ in common deck COMSACC) are considered special user names. Refer to the NOS Version 2 Administration Handbook for further information on VALINDS and special user names.
SSPMN	70 ₈	Minimum security system prolog index.

The NOS Version 2 Administration Handbook describes the use of the following COMSACC user control parameters.

Parameter	Default	Description
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 Version 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.
KCCI	100B	Default limit for commands processed; the maximum value is 176 ₈ .
KCMI	37B	Default limit for central memory field length/100 ₈ ; the maximum value is 37 ₈ .
KCPI	0	Default limit for cards punched from a file; the maximum value is 76 ₈ .
KDFI	100B	Default limit for dayfile messages written; the maximum value is 176 ₈ .
KDTI	0	Default limit for the number of detached jobs; the maximum value is 37 ₈ .
KECI	0	Default limit for extended memory field length/1000 ₈ ; the maximum value is 176 ₈ .
KLPI	1000B	Default limit for lines printed from a file; the maximum value is 3776 ₈ .

Parameter	Default	Description
KMSI	1000B	Default limit for additionally allocated mass storage PRUs; the maximum value is 7776 ₈ .
KPTI	1000B	Default limit for the number of units plotted; the maximum value is 76000 ₈ .
KSLI	10B	Default limit for SRU accumulation; the maximum value is 76 ₈ .
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 Version 2 Administration Handbook.

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	Description
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	Description
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 supported devices. Assemble CALLSYS to obtain a listing of COMSJIO.

Parameter	Default	Description
LR\$	Defined	Any 580-12 printer.
LS\$	Defined	Any 580-16 printer.
LT\$	Defined	Any 580-20 printer.
LX\$	Defined	Any 5870 nonimpact printer.
LY\$	Defined	Any 5970 nonimpact printer.
P8\$	Defined	Punch 80-column binary.
PB\$	Defined	Punch system binary.
PH\$	Defined	Punch coded.
PL\$	Defined	Plotter.
PR\$	Defined	Any line printer.
PU\$	Defined	Punch coded.
SB\$	Defined	Punch system binary.
WT\$	Defined	Wait Queue.

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	Description
LTKL	20B	<p>If you did not initialize a mass storage device during deadstart (using the INITIALIZE entry described in the NOS Version 2 Analysis Handbook), the system searches the device for a label that might be in track 0.</p> <p>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, 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 Version 2 Analysis Handbook.</p>

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. The site security administrator supplies any changes to be made to these micros. Assemble CALLSYS to obtain a listing of COMSMLS.

Parameter	Default	Description
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.

COMSMMF Parameters

COMSMMF contains parameters that define the multimainframe tables that reside on the link device. Assemble CALLSYS to obtain a listing of COMSMMF.

Parameter	Default	Description
MXMF	4	Maximum number of mainframes in a multimainframe configuration. Two to seven mainframes are allowed.

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	Description
AFDL	20000B	Account file threshold size in PRUs. ³
BLTL	1000B	Binary maintenance log threshold size in PRUs. ³
DFDL	20000B	Dayfile threshold size in PRUs. ³
ELDL	1000B	Error log threshold size in PRUs. ³
HRTL	2	Maximum number of times + 1 that a job will be rerun due to a hardware error.
MSER	2	Maximum number of times + 1 that a job will be rerun due to a software error.
MXSY	5	Maximum number of devices that can be defined as system devices during deadstart.
SYSCHG	SYSTEM	Specifies the system charge number for jobs initiated under DSD.
UPTL	10B	Maximum number of uncorrected processor errors that can occur per hour before the operator is notified.

³. When entries in any of these files reach the threshold, the A,OPERATOR flashing message appears on the console screen (refer to the NOS Version 2 Operations 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	Description
MUNIT	16D	Maximum number of tape units defined per mainframe. MUNIT can be set to any value up to 33D.
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.
POGH	0	<p>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 Version 2 Reference Set, Volume 3).</p> <p>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.</p> <p>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.</p>
POLM	0	<p>Flag indicating whether tape detailed status error messages are issued to the job dayfile. If POLM is 0, the system does not issue these messages to the job dayfile. If POLM is 1, the system issues the message to the dayfile for both the first and the last attempt to read a bad tape block. The user can override the installation setting of POLM with parameters on the tape assignment command (refer to the NOS Version 2 Reference Set, Volume 3).</p> <p>The system issues all tape error messages to the error log regardless of the setting of POLM.</p>

Parameter	Default	Description
PONR	0	<p>Flag indicating whether the system automatically assigns a mounted tape that cannot be verified as the tape being requested. This pertains only to the following situations:</p> <ol style="list-style-type: none"> 1. Mounting a second or later reel in a multi-reel unlabeled tape request. 2. Mounting a second or later reel in a multi-reel labeled tape request where no VSN was specified on the request. 3. Mounting a first reel in an unlabeled tape request after a ring conflict has occurred on that reel. <p>If PONR is 0, the system does the following:</p> <ul style="list-style-type: none"> • In situations 1 and 2, the E, P display issues the message MOUNT to inform the operator to mount (or if the tape is already mounted on another drive, to assign via the VSN command) the next tape. Any tape subsequently mounted or assigned is then automatically accepted. • In situation 3, any tape subsequently mounted with the correct ring status is automatically accepted. <p>If PONR is 1, the system does the following:</p> <ul style="list-style-type: none"> • In situations 1 and 2, the E, P display issues the message CHECK AND MOUNT to inform the operator to visually verify that the next tape to be mounted or assigned is in fact the tape being requested. The system then gives the operator the option of either dropping the job (if the correct tape could not be found) or entering the DSD command, NEXTREEL,est., to inform the system that the operator has found the correct tape. Any tape subsequently mounted or assigned is then automatically accepted. If the NEXTREEL command is not entered before the next tape is mounted or assigned, the system rejects and unloads the tape and redisplay the CHECK AND MOUNT message. • In situation 3, the operator must enter the NEXTREEL command before remounting the tape with the correct ring status. Failure to do so causes the system to reject and unload the tape and display the CHECK AND MOUNT message.

Parameter	Default	Description
SRRM	2	Stage request retry maximum. This is the number of times that an unsuccessful attempt to stage a file from tape alternate storage is to be retried before the stage request is abandoned. SRRM can be set to values franging from 0 to 7; a value of 7 indicates that stage requests should be retried indefinitely.
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	Description										
ACDF	ACNO	Alternate user CATLIST permission (AC) default specification for new files; ACDF can be set to the following values: <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ACNO</td> <td>Newly created files are not CATLISTable.</td> </tr> <tr> <td>ACYS</td> <td>Newly created files are CATLISTable.</td> </tr> </tbody> </table>	Value	Description	ACNO	Newly created files are not CATLISTable.	ACYS	Newly created files are CATLISTable.				
Value	Description											
ACNO	Newly created files are not CATLISTable.											
ACYS	Newly created files are CATLISTable.											
ACEX	ACNO	Alternate user CATLIST permission (AC) default specification for existing files (files created prior to NOS level 617); ACEX can be set to the following symbolic values: <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ACNO</td> <td>Existing files are not CATLISTable.</td> </tr> <tr> <td>ACYS</td> <td>Existing files are CATLISTable.</td> </tr> </tbody> </table>	Value	Description	ACNO	Existing files are not CATLISTable.	ACYS	Existing files are CATLISTable.				
Value	Description											
ACNO	Existing files are not CATLISTable.											
ACYS	Existing files are CATLISTable.											
APLO	1	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 for a temporary deadlock exists in the latter instance. If this happens, the RHF applications involved are timed-out and aborted.										
BRDE	BRAL	Backup requirement (BR) default specifications; can be set to the following symbolic values. <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>BRAL</td> <td>Backup always required.</td> </tr> <tr> <td>BRMD</td> <td>Media-dependent backup for systems with a supplemental mass storage facility.</td> </tr> <tr> <td>BRNO</td> <td>No backup required.</td> </tr> <tr> <td>GRMX</td> <td>Maximum BR value.</td> </tr> </tbody> </table>	Value	Description	BRAL	Backup always required.	BRMD	Media-dependent backup for systems with a supplemental mass storage facility.	BRNO	No backup required.	GRMX	Maximum BR value.
Value	Description											
BRAL	Backup always required.											
BRMD	Media-dependent backup for systems with a supplemental mass storage facility.											
BRNO	No backup required.											
GRMX	Maximum BR value.											

Parameter	Default	Description
DFPT	DI1	Equipment type. When accessing a removable 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. Changes in the default may be made through the DFPT IPRDECK entry.
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 the 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 the site security administrator.
MNHS	5	<p>Minimum size hole, in sectors, that the permanent file manager (PFM) creates in the indirect access file chain when using an existing hole. If, in searching for a hole in which to save an indirect access file, PFM finds that using 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. Purging a file whose total length is less than MNHS creates a hole smaller than MNHS.</p> <p>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.</p> <p>The minimum value for MNHS is 3; the maximum value is 7777₈.</p>

Parameter	Default	Description
PGUI	300000 ₈	PFDUMP (refer to the NOS Version 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.
PMLM	62 ₁₀	Number of explicit permissions allowed per file, 1 through 4095. If PMLM is changed, PFM must be reassembled.
RSDE	RSNP	Preferred residence (PR) default specification; can be set to the following symbolic values.
	Value	Description
	RSDS	Disk residence preferred.
	RSLK	Locked to disk.
	RSMS	Mass storage facility residence preferred.
	RSMX	Maximum PR value.
	RSNP	No preferred residence.

For individual users, each of four permanent file access limits is established through MODVAL (refer to the NOS Version 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 7-9 summarizes the released values for each parameter. Setting a parameter to 0 indicates unlimited access.

Parameter	Description
CSRNGn	Upper limit of range n for cumulative size of indirect access files, specified in PRUs; must not exceed 777777 ₈ .
DSRNGn	Upper limit of range n for size of individual direct 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 .
NFRNGn	Upper limit of range n for file count; must not exceed 77777 ₈ .

Table 7-9. Released Values of Permanent File Limit Ranges

Parameter	n=1 ¹	n=2 ¹	n=3 ¹	n=4 ¹	n=5 ¹	n=6 ¹	n=7 ¹
CSRNGn	1000	2000	5000	10000	50000	100000	0
DSRNGn	1000	2000	5000	10000	50000	100000	0
FSRNGn	10	30	50	100	150	300	0
NFRNGn	10	20	30	40	50	100	0

1. 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	Description
PPFN	PROFILC	Micro definition specifying the PROFILE routine's database file name (refer to the NOS Version 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	Description
NMFL	60000B	Defines the size of NAM's field length as used by the algorithm in COMPCM. This calculation determines the field length available for an interactive job. This value should be equal to the value of MAXFL, which defines the maximum field length that NAM can attain. MAXFL is a parameter on the NIP command.
TAPC	20B	Number of pots of typed ahead input to be stored in IAF's FL for each user.
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 Version 2 Reference Set, Volume 3). For details of the algorithm used in calculating the default time limit, refer to the NOS Version 2 Administration Handbook.
VDSI	100B	Default system resource unit (SRU) and time limit
VDTI	100B	increment values for the S,nnnnn and T,nnnnn interactive commands.
VNCP	40B	Number of pots of output to be stored in IAF's FL for each user.
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.

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	Description
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. <ul style="list-style-type: none"> • 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).
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.
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.
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.

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	Description
FLLM	50000 _g	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 Version 2 Administration 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	Description
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.

COMSVED Parameters

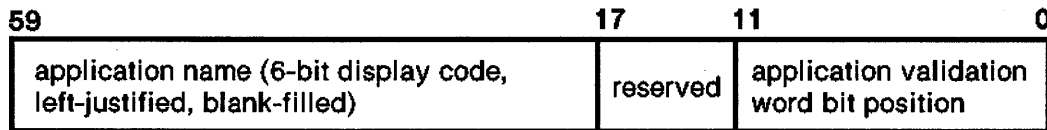
COMSVED defines two assembly constants to determine how many pool and driver PPs to reserve for NOS use in a dual state environment. Assemble CALLSYS to obtain a listing of COMSVED.

Parameter	Default	Description
MINDP	MINP/2	Defines the number of driver PPs reserved for NOS on a CYBER 180-810/830 with 20 PPs.
MINP	4	Defines the number of pool PPs reserved for NOS.

The number of PPs reserved by MINP and MINDP is in addition to dedicated PPs (such as MTR and DSD) and PP programs which occupy a PP for a relatively long time (such as 1CD). Note that this code is executed by VER (assembled in the DUAL installation procedure).

COMTNAP Parameters

COMTNAP defines a table that maps valid NAM 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 Version 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.



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:

Bit Position	Application Name
0	IAF
1	RBF
2	TAF
3	MCS
4	TVF
5	CS
6	PLATO
7	ITF
8	TLF
9	NJF
10	NETOU
11	PSU
12	AP1
13	AP2
14	AP3
15	VEIAF
16	NPF
17	TCF
18	AP4
19	AP5
20	AP6

The following table-related parameters are defined in COMTNAP. All symbols are unqualified.

Parameter	Default	Description
TNAV	-	Table first word address. Program that uses COMTNAP defines the value.
TNAVL	25 ₈	Table length, excluding zero-word terminator.

PPCOM Parameters

PPCOM contains the following parameters used by system peripheral processor packages for intercommunication. Assemble PPTEXT to obtain a listing of PPCOM.

Parameter	Default	Description
CCFL	224B	Initial field length of 100B assigned to CCLBRWE for prolog processing. Changes to CCL installation parameters may require adjustment of the value of this symbol.
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.
MXLF	128D	Maximum number of local files. This value affects the maximum negative field length (see MNFL in PPCOM) and the maximum central memory field length available for a job. Both are calculated in multiples of 100B. The MXLF default of 128D gives a maximum negative field length of 1200B and a maximum user CM field length of 376500B. Note that increasing MXLF (and thus increasing MNFL) reduces the amount of CM field length available for all user jobs.

The assembly constant INSP\$ is defined in PPCOM. If the INSP\$ reference is deleted, 10 bytes become available for site code in both the DSD display and the command overlays.

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	Description
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.
SDF	–	When specified in an ENTER macro call, the associated command is logged in the system dayfile.

819 PPU Driver Installation

You can install the 819 PPU driver on the model 176 computer only. 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. If you have a new or updated version of this microcode, the deadstart tape must be updated to contain it. To build the 819 driver, execute the following command:

```
BEGIN, HCD, INSTALL.
```

Binaries from this job are copied to a permanent file named PRODUCT.

NSS - NOS Scope 2 Station Facility 1

There are no special build instructions for NSS. For information about initiating SSF, refer to Subsystem Initiation at the beginning of this chapter. This section describes the following:

- Configuration Information
- Installation Procedure Messages
- Spun-Off Task (SPOT) Memory Requirements
- Installation Parameters

Configuration Information

To configure the NOS Scope Station Facility, create the following:

- **SSPOT user name.** Create user name SSPOT for running the SSF subsystem. (The user name is created for you automatically in a first-time installation.) The password for this user name is (by default) STATION, but should be changed after installation is complete. The station has been tested with the following validations, some of which may not be necessary.

MT=7	DB=7
RP=7	LP=77B
SC=50B	CC=77B
MS=77B	DF=77B
CM=20B	SL=77B
TL=77B	AW=CLPF, CSPF, CCNR, CASF, CAND, CSR, CSAP
CP=77B	

To change the user name or password, create a USER file which applies a modification set to change line NSSA000.4601 then run the NSS installation procedure. (File ZZSYSGU should also be updated to reflect the new user name/password. Refer to SYSGEN Validations in chapter 8 for more information.)

- **LIDCMid file.** Entries in this file define the physical and logical machines available to your system. (The id in LIDCMid is the two alphanumeric characters from the MID CMRDECK entry which make up your machine identifier. The released default id is AA.) The number of entries in the LIDCMid file determine the value specified in the LDT CMRDECK entry. The LIDCMid file is stored as an indirect access permanent file on user name SYSTEMX (user index 377777B). In addition to entries in this file for NSS, entries are also made for dual state, RHF, and PTF/QTF.

If you are installing dual state, a sample LIDCMid file is provided with the release materials. To examine this file, consult the DUAL section of this chapter. If you are not installing dual state, create the LIDCMid file on the installation user name using an available text editor:

To move the LIDCMid file to user name SYSTEMX, execute the following command from the system console when directed by the instructions in chapters 2, 3, or 4:

```
SYSGEN(MOVE,LIDCMid,SYSTEMX,,Y,PERMIT)
```

(The PERMIT parameter allows read access to the file from the installation user name.) Once the file has been moved, build the LID table in central memory using the CLDT system program. To use CLDT, ensure that NAM, IAF, RHF, and SSF are not running and enter the following command from the system console:

```
X.CLDT.
```

CLDT is executed automatically at NOS deadstart time if an LDT entry exists in the CMRDECK.

- LDT CMRDECK entry. This entry defines the size of the Logical Identifier Table in central memory. The size of the table is determined by the number of entries in the LIDCMid file.
- CC EQPDECK entry. This entry defines the connection of each 6683 Satellite coupler to your mainframe.

For more information about the LIDCMid file and LDT CMRDECK entry, refer to the LID/RHF Configuration Files section of the NOS Version 2 Analysis Handbook. Information about the CC EQPDECK entry plus additional information about the LDT CMRDECK entry can be found in the Deadstart Decks section of that manual.

Installation Procedure Messages

The DECKOPL installation procedure for NSS contains two errors. The following messages appear in the dayfile:

```
COPYL DID NOT FIND -- REL      / MSC
COPYL DID NOT FIND -- REL      / SEC
```

This condition is non-fatal and does not affect the generated binaries. The frequency of occurrence is relative to this product as released; any local code may change the frequency.

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
LBUFDP	4001	7000 dump
LBUFDSLM	7725	Deadstart link buffer
LBUFDSTP	3004	Deadstart via tape
LBUFLM	1401	Link medium
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 7-10.

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 7-10. SPOT Octal Memory Requirements

SPOT Type	Code	Buffer Lengths Used to Transfer Files	Total Memory Used	Notes¹
Deadstart	6200	LBUFDSTP+ LBUFDSLM	21200	Via tape
Dump	4200	LBUFDP	10200	
Permanent file	5100	LBUFPF76+ LBUFLM	11600	
Spooling	10200	LBUFSP+ LBUFLM	14600	Single file transfer
Tape staging ²	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]

1. Figures in brackets are the results using the released default values for the parameter variables in the equations.

2. 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.

Installation Parameters

The following installation parameters are all on common deck COMTUNE on the NSS program library. Deck COMTUNE also lists the minimum and maximum values for these parameters.

Table Size Parameters

Parameter	Default	Description
DATAENT	0	The maximum number of active data streams.
DISNAMESIZE	10	The size of the display name table for the transparent display interface to the Scope 2 operating system.
IDTMAX	10	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. MAXSPOTS refers to a group of parameters; therefore, assemble deck COMTUNE for a list of default values.
NMF	1	The number of mainframes to which the station can be linked; the maximum value that can be specified is 2.
SNTS	11	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).
SPOOLS	4	The maximum number of spooling streams.
SYNTAXSIZE	200	The size of the syntax extension table for the transparent display interface to the Scope 2 operating system.

Spooling and Recalling Time Parameters

Parameter	Description
BSYLIM	The length of time in seconds the busy overlay of MFSTAT delays after sensing an idle condition, before going into an idle state.
DSDWAIT	The length of time in seconds that MFSTAT waits for a reply before it rejects a DSD request.
IDLEMAX	The elapsed time in seconds that MFSTAT waits after all spooling activity has completed before swapping out the spooling SPOT.
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.
IDLRCLTM	The recall time in seconds used by MFSTAT when the idle overlay is executing.
ISEC(i)	A way of approximating the number of idle station loops in i seconds.
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).
MAXINCOUNT	The frequency in seconds with which MFSTAT checks the input queues for files to spool when it is idle.
MSEC(i)	Recall time in milliseconds. For example, MSEC(5) is equal to 5 milliseconds.
MSGCNT	The length of time in seconds that MFSTAT leaves informative messages on the B display.
OVLMAX	The maximum time in seconds that MFSTAT retains the secondary overlay field length after a load of a secondary overlay.
RCL7000	The recall time in milliseconds used by the busy overlay of MFSTAT when communicating with a linked Scope 2 operating system.
SEC(i)	A method of approximating the number of busy station loops in i seconds.
SPLLIM	The number of seconds MFSTAT waits before going idle once spooling activity is completed.

Parameter	Description
SPOOLRCL	The recall time in milliseconds used by the spooling SPOT when there is no spooling activity.
STA7000	The delay in seconds used by MFSTAT between sending status requests to the Scope 2 operating system.
TIMOUT	The length of time in seconds that MFSTAT waits before logging out a linked mainframe when communication is lost.
TME7000	The delay in seconds between sending time requests to the Scope 2 operating system.

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.

Buffer Size Parameters

Parameter	Default	Description
DAYBUFSIZE	220	The size of the MFSTAT buffer for processing SPOT dayfiles on the active side.
LBUFLM	769	The length of the link buffer used by SPOT jobs for NOS Scope 2 I/O spooling.
LBUFPF76	2601	The length of the disk buffer used to read and write the disk for permanent file transfers to and from Scope 2.
LBUFSP	1041	The length of the disk buffer used by the spooling SPOT for NOS Scope 2 spooling of I/O files.
LICRBUF	76B	The length of the MFSTAT buffer used by the IAF queue utility helper.
LRGBUF	440B	The size of the MFSTAT receive buffer for the active side of the station.
RRBUF	15B	The size of the MFSTAT active transmit buffer and the linked staged packet buffer.

The following installation parameter is in the deck HELLO7:

Parameter	Default	Description
NOMFF	1	If NOMFF is set to 1, HELLO7 supports a single mainframe. For multiple mainframe support, NOMFF must be set to 0.

PSU - Printer Support Utility Version 1.1

There may be up to twelve printers connected to PSU, consisting of any mixture of 533, 536, 537, 585, or Apple LaserWriter printers. If the printer's configuration is not correctly described in the default PSU EVFULFN file, changes to the file are required.

This section describes the following:

- Configuration Information
- Unique Parameters
- PSU Command
- File Placement
- NDL File Entries

Configuration Information

To configure the Printer Support Utility, create the following:

- Electronic Vertical Format Unit Load File (EVFULFN). This file sets default options for PSU and is stored as a public indirect access permanent file on the network operations user name. The file contains legible text and may be modified in advance using the instructions below.

For a first-time installation, GET EVFULFN from the network operations user name directly. If you are installing PSU for the first time during a NOS upgrade or as part of a component order, execute the SYSGEN(PSU1) procedure from the installation user name to load a copy of EVFULFN and the PSU NAMSTRT records to the installation user name (file NAMSTRT is required). To move the EVFULFN file to the network operations user name, execute the following command from the system console:

```
SYSGEN(MOVE,EVFULFN,netopun,,Y,PERMIT)
```

Refer to the instructions in chapter 2, 3, or 4. (The PERMIT parameter allows read access to the file from the installation user name.)

- Validation file entries. PSU requires user names validated for the PSU application to print files. User names are of the form PRINTnn, where nn is a 2-digit number. These user names are referenced in the EVFULFN file. SYSGEN creates user names PRINT01 through PRINT12.
- PCFid file. If you are installing PSU as part of an upgrade or component order, you must purge file PCFid from user name SYSTEMX. The id in PCFid is the 2-character alphanumeric identifier from your MID CMRDECK entry.

To configure a printer that is connected to a 255x NPU, create the following:

- NDL file entries. Entries are required in the Network Configuration File (NCF) and Local Configuration File (LCF) portions of a Network Definition Language (NDL) file. A LINE statement in the NCF is required to define the connection between the NPU and the printer and a USER statement in the LCF is required to automatically connect and log the printer into the system. The user name specified on the USER statement is used by PSU to determine the class (attributes) of the printer. This association is made in the EVFULFN file.

A sample NDL file containing the required entries is provided with the release materials. To examine this file, consult the NAM5 section of this chapter. In addition, the NAM5 section contains information on how to produce the LCFFILE and NCFFILE using the Network Definition Language Processor (NDLP) to compile the NDL file. Refer to NDL File Entries in this section for more information.

- EVFULFN file entries. The entries in this file define the class (characteristics) of the printer that uses each PRINTnn user name.

If the printer is a 533 or 536, use user names PRINT01-PRINT04 or change EVFULFN to use a user name with printer class C536INT. All user names using printer class C536INT must be defined last (i.e., these user names must follow all user names that use some other printer class).

If the printer is a 537 or other asynchronous printer, use user names PRINT05-PRINT06 or change EVFULFN to use a user name with printer class INTNVFU (Interactive with No VFU).

Note the CDC 585 URI printer is only supported by CDCNET.

To configure a printer that is connected to a CDCNET MTI or TDI, create the following:

- NDL file entries. Entries are required in the Local Configuration File (LCF) portion of a Network Definition Language (NDL) file. A USER statement in the LCF is required to automatically connect and log the printer into the system. The user name specified on the USER statement is used by PSU to determine the class (attributes) of the printer. This association is made in the EVFULFN file.

A sample NDL file containing the required entries is provided with the release materials. To examine this file, consult the NAM5 section of this chapter. In addition, the NAM5 section contains information on how to produce the LCFFILE and NCFFILE using the Network Definition Language Processor (NDLP) to compile the NDL file. Refer to NDL File Entries in this section for more information.

- CDCNET Configuration File entries. A DEFINE_LINE statement must appear in the system configuration file for the DI to define the connection between the DI and the printer. A terminal definition procedure (TDP) is also required for proper configuration of the printer. Where possible, it is desirable to create a VFU Load Procedure for your printer. These procedures may be defined for CDC 533, 536, 537, and 585 (URI) printers. An Apple LaserWriter may not have a VFU Load Procedure but must have a File Prefix Procedure.

Sample CDCNET configuration files are provided with the release materials. Refer to the CDCNET Configuration Guide for a description of these files.

- EVFULFN file entries. The entries in this file define the class (characteristics) of the printer that uses each PRINTnn user name.

If the printer has a defined CDCNET VFU Load Procedure, use user names PRINT09-PRINT12 or change EVFULFN to use a user name with printer class BATWVFU (Batch with VFU). This could include the CDC 533, 536, 537, and 585 (or other URI printers) printers.

If the printer does not have a defined CDCNET VFU Load Procedure, use user names PRINT07-PRINT08 or change EVFULFN to use a user name with printer class BATNVFU (Batch with No VFU). This would include an Apple LaserWriter printer.

For more information about PSU, consult the PSU section in the NOS Version 2 Analysis Handbook.

Unique Parameters

Parameter	Description
-----------	-------------

NOTRACE	To disable AIP trace file creation, specify the keyword NOTRACE on the call to the PSU installation procedure.
---------	--

PSU Command

The following command initiates PSU:

```
PSU.
```

The PSU command is in the PSU startup procedure file, which is part of the NAMSTRT file. Thus, PSU is started automatically with the network.

File Placement

There are two files that must be moved for PSU: the NAM startup master file NAMSTRT and the electronic vertical format unit load file EVFULFN.

If you rebuild NAM, execute the command SYSGEN(PSU1) from an interactive terminal logged in to your installation user name to add the PSU record to NAMSTRT. SYSGEN(PSU1) requests your released permanent file tapes and loads the file PFGPSU1. This file contains the released NAMSTRT records and a sample EVFULFN. The EVFULFN file must be installed to the network operations user name. For more information about the EVFULFN file, refer to the NOS Version 2 Analysis Handbook.

You can move the NAMSTRT and EVFULFN files to the appropriate user names by using the SYSGEN(MOVE) command (refer to chapter 8).

NDL File Entries

For each printer connected to PSU on a 255x NPU, you must include the following Network Definition Language (NDL) statements:

- A LINE statement in the Network Configuration File (NCF) with the following format:

```
linex:  LINE,PORT=portx,LTYPE=A2,TIPTYPE=ASYNC,LSPEED=9600.
```

- A TERMDEV statement in the NCF with the following format:

```
devicex:  TERMDEV,TC=721,AUTOCON,HN=hostnode,LK=YES,OC=YES,PA=0,PW=0,
          PL=0,ABL=3,DBL=2.
```

- A USER statement in the Local Configuration File (LCF) with the following format:

```
devicex:  USER,MFAM=family,MUSER=PRINTnn,MAPPL=PSU.
```

Parameter	Description
devicex	Device name for a PSU printer.
family	Name of the family containing user name PRINTnn, where nn is a 2-digit number. User name PRINTnn must be validated to use PSU.
hostnode	Node number of the host coupler to which the PSU printer is connected.
linex	Line name for a PSU printer.
portx	Port number for a PSU printer line.

The released NDL file NDLDATA (located on the network administration user name) contains definitions for two PSU printers.

QU3 - Query Update Version 3

The installation tool SYNGEN, and the various common decks that reside on the DDL3 program library, must be available for the installation of Query Update 3.

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. The QU3 installation procedure contains commands to access this file and to create a temporary library, DBULIB, to be used at load time. The procedure 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.

NOTE

QU3 has several installation options that are affected by changing parameters in common decks NUMOPT and TOPTION. Refer to QU3's program library for detailed explanations of all QU3 installation options.

RBF5/RBF5D - Remote Batch Facility Version 1

This section describes the following installation options for RBF5:

- Special Notes
- Unique Parameters
- RBF Command
- USER File Directives
- File Placement

Special Notes

RBF5 binaries are released on the deadstart tape in non-debug/trace mode. Control Data provides debug/trace binaries of RBF5 on the permanent file tapes. To obtain these binaries, use the SYSGEN(SWAP) function as described in chapter 8.

Unique Parameters

Parameter	Description
-----------	-------------

NOTRACE	To disable log file creation for RBF5, specify the keyword NOTRACE on the call that executes the RBF5 installation procedure. (This option is not available for RBF5D.)
---------	---

RBF Command

RBF5 requires the following command:

```
RBF2P0 (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. By default, the MC parameter picks up its value from NAMSTRT. 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, 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
```

in a USER file for the RBF5 installation procedure.

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 a USER file for the RBF5 installation procedure.

Parameter	Default (decimal)	Description
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.
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.
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.
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.
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.
STATIONS	16	Maximum number of consoles.
TOTDEV	32	Maximum number of batch devices.

File Placement

Execution of the RBF5 installation procedure requires a prior build of NAM and the existence of the NAM startup master file on the installation user name. The RBF5 installation procedure modifies the NAM startup master file (NAMSTRT). For more information about the network startup master file, refer to the NAM5 section of this chapter.

RDFEX - Remote Diagnostic Facility

RDFEX and IAF share decks IAFEX and 1TM in the composite OPL (OPLpsrout). Both products must be rebuilt if user code is applied to either of these decks. That is, user code must be included for both the RDFEX and IAF build procedures.

RHF - Remote Host Facility Version 1

This section describes the following installation options for RHF:

- Configuration Information
- RHF Procedure File
- Hardware Configuration
- Installation Parameters
- NAD Microcode

Configuration Information

To configure the Remote Host Facility, create the following:

- **RCFMid File.** Entries in this file define loosely coupled network (LCN) elements. These include Network Access Devices (NADs), applications, and mainframe physical identifiers (PIDs) for all LCN configurations used by or accessible to RHF on your mainframe. (The id in RCFMid is the two alphanumeric characters from the MID CMRDECK entry which make up your machine identifier. The released default id is AA.) The RCFMid file is stored as a direct access permanent file on user name SYSTEMX (user index 377777B). This file should be created or edited on the installation user name then moved to user name SYSTEMX.

To create the RCFMid file, first make a file containing network configuration statements on the installation user name. This file is then processed by the system utility RCFGGEN to produce RCFMid. An example sequence of commands is given below:

1. Create RCFGGEN input using an available text editor. (This example uses the file name rcfin.)
2. If you are installing RHF as part of a system upgrade or a component order (and have therefore not deadstarted the system containing the new level of RCFGGEN), access the new version from GLOBLIB:

```
ATTACH,GLOBLIB.  
LIBRARY,GLOBLIB/A.
```

3. Execute RCFGGEN:

```
RETURN,RCFMid.  
PURGE,RCFMid. (Be sure to PURGE any previous file.)  
DEFINE,RCFMid.  
REPLACE,rcfin.  
RCFGGEN,I=rcfin,L=list,O=RCFMid.  
RETURN,RCFMid.
```

File list will contain any diagnostic messages from RCFGGEN.

To move the RCFMid file to user name SYSTEMX, execute the following command from the system console when directed to do so in chapter 2, 3, or 4:

```
SYSGEN,MOVE,RCFMid,SYSTEMX.
```


- LIDCMid file. Entries in this file define the physical and logical machines available to your system. (The id in LIDCMid is the 2-character alphanumeric identifier from the MID CMRDECK entry. The released default id is AA.) The number of entries in the LIDCMid file determine the value specified in the LDT CMRDECK entry. The LIDCMid file is stored as an indirect access permanent file on user name SYSTEMX (user index 377777B). In addition to entries in this file for RHF, entries are also made for dual state, Remote Host Products (PTF/QTF), and the NOS Scope Station Facility.

If you are installing dual state, a sample LIDCMid file is provided with the release materials. To examine this file, consult the DUAL section of this chapter. If you are not installing dual state, create the LIDCMid file on the installation user name using an available text editor.

To move the LIDCMid file to user name SYSTEMX, execute the following command from the system console when directed to do so in chapter 2, 3, or 4:

```
SYSGEN,MOVE,LIDCMid,SYSTEMX,,Y,PERMIT.
```

(The PERMIT parameter allows read access to the file from the installation user name.) Once the file has been moved, build the LID table in central memory using the CLDT system program. To use CLDT, ensure that IAF, NAM, RHF, and SSF are not running and enter the following command from the system console:

```
X.CLDT.
```

(CLDT is executed automatically at NOS deadstart time if an LDT entry exists in the CMRDECK.)

- LDT CMRDECK entry. This entry defines the size of the Logical Identifier Table in central memory. The size of the table is determined by the number of entries in the LIDCMid file.
- NC EQPDECK entry. This entry defines the connection of each Network Access Device (NAD) to your mainframe.

For more information about the RCFMid file, LIDCMid file, and LDT CMRDECK entry, refer to the LID/RHF Configuration Files section of the NOS Version 2 Analysis Handbook. Information about the NC EQPDECK entry and additional information about the LDT CMRDECK entry can be found in the Deadstart Decks section of that manual.

RHF Procedure File

The entry point of the RHF subsystem is named RHF. Therefore, you can initiate RHF without a procedure file with the DSD entry RHF.

If you want to initiate RHF with a procedure file, create an RHF procedure file. If you create such a file, it 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 SYSTEMX (user index 377777₈).

Refer to the beginning of this chapter 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).

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 Network Access Device Hardware Reference Manual for further information.

Installation Parameters

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:												
	<pre> 1 7 22 DEF FETBUFSIZE #3200#; </pre> <table> <thead> <tr> <th></th> <th>Assigned (binary)</th> <th>Assigned (coded)</th> </tr> </thead> <tbody> <tr> <td>0 to 532</td> <td>532</td> <td>992</td> </tr> <tr> <td>532 to 992</td> <td>532 to 992</td> <td>992</td> </tr> <tr> <td>992 and up</td> <td>992 and up</td> <td>992 and up</td> </tr> </tbody> </table>		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
	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#;
```

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#;
```

NAD Microcode Initialization Parameters

A set of initialization parameters must be loaded into NAD memory along with the NAD microcode as documented in the 380-170 Network Access Device Hardware Reference Manual. These parameters are compiled into MHF and are appended to all NAD microcode 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 RHF. MHF 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 microcode memory use.

NAD buffer tracing can be controlled when generating the RHF configuration file. Refer to the TRACE parameter on the LNAD statement, described in the LID/RHF Configuration Files chapter of the NOS Version 2 Analysis Handbook.

NAD Microcode

NAD microcode resides on the system as a PPU-type record named 170. You can load NAD microcode during RHF initialization or by operator request.

To build NAD microcode, submit the following job. The binaries produced by this job are copied to permanent file PRODUCT.

```
job command.
USER,username,password,familyname.
LABEL,TAPE,F=fmt,tapetyp,D=density,LB=KU.
NOTE,IN.>*COMMENT PPU/170 170 FIRMWARE NOS LVL-xx
COPYBF,TAPE,INHOLD.
BEGIN,170,INSTALL.
---eoi---
```

Parameter	Description
density	Tape density. For example, PE or GE.
fmt	Format of tape. For example, I or SI.
tapetyp	Tape type. For example, MT or NT.
xx	Version level. For example, 05.

RHP - PTF/QTF File Transfer Facility

This section describes the following options for RHP:

- Configuration Information
- Unique Parameters
- Installation Parameters
- QTF Initiation Procedure Parameters
- QTF Configuration Directives

Configuration Information

To configure the PTF/QTF File Transfer Facility, create the following:

- RCFMid file. Entries in this file allow PTF/QTF file transfers to take place using an RHF LCN network. The entries define the PTF/QTF applications (APPL statements) and the interhost and intrahost application-to-application connections (NPID, LNAD, RNAD, and PATH statements). Refer to the RHF section of this chapter for more information.
- LIDCMid file. Entries in this file define the physical and logical machines available to your system. (The id in LIDCMid is the two alphanumeric characters in the MID CMRDECK entry. The released default id is AA.) The number of entries in the LIDCMid file determine the value specified in the LDT CMRDECK entry. The LIDCMid file is stored as an indirect access permanent file on user name SYSTEMX (user index 377777B). In addition to entries in this file for PTF/QTF, entries are also made for dual state, the Remote Host Facility (RHF), and the NOS Scope Station Facility.

If you are installing dual state, a sample LIDCMid file is provided with the release materials. To examine this file, refer to the DUAL section of this chapter. If you are not installing dual state, create the LIDCMid file on the installation user name using an available text editor.

To move the LIDCMid file to user name SYSTEMX, execute the following command:

```
X.SYSGEN(MOVE,LIDCMid,SYSTEMX,,Y,PERMIT)
```

from the system console when directed to by instructions in chapters 2, 3, or 4. (The PERMIT parameter will allow read access to the file from the installation user name.) Once the file has been moved, build the LID table in central memory using the CLDT system program. To use CLDT, ensure that IAF, NAM, RHF, and SSF are not running and enter the following command from the system console:

```
X.CLDT.
```

(CLDT is executed automatically at NOS deadstart time if an LDT entry exists in the CMRDECK.)

- LDT CMRDECK entry. This entry defines the size of the Logical Identifier Table in central memory. The size of the table is determined by the number of entries in the LIDCMid file.

- NDL file entries. Entries in the Local Configuration File (LCF) portion of a Network Definition Language (NDL) file allow PTF/QTF file transfers to take place using a NAM/CCP network or a NAM/CDCNET network. The entries define the PTF/QTF applications (APPL statements) and the interhost and intrahost application-to-application connections (INCALL and OUTCALL statements).

Transfers using a NAM/CCP network require entries in the Network Configuration File (NCF) portion of the NDL file. For non-X.25 transfers, LOGLINK statements (and possibly TRUNK statements) define host-to-host logical links. For X.25 transfers, LINE statements define the network's X.25 connection. The NPU that contains the X.25 line must use a variant containing the X.25 and X.25 A-A TIPs (referenced by the NPU statements VARIANT parameter).

Transfers using a NAM/CDCNET network require the network products gateway to CDCNET to be defined using the DEFINE_NP_GW command in the configuration file for the Mainframe Device Interface (MDI) that contains the X.25 line.

A sample NDL file containing PTF/QTF application definitions is provided with the release materials. To examine this file, refer to the NAM5 section of this chapter. Also in the NAM5 section is information on how to produce the NCFFILE and LCFFILE using the Network Definition Language Processor (NDLP) to compile the NDL file.

- NLFFILE variant. X.25 transfers using a NAM/CCP network require that the NPU with the X.25 line use an NPU variant which includes the X.25 and X.25 A-A TIPs. The NPU load file variant is contained in the Network Load File, NLFFILE. This variant is referenced from the VARIANT parameter on a NPU statement in the NDL file.

If CCP option A or B was ordered from the OIP, the released CCP load file contains a variant defining X.25. For a complete list of attributes for the released CCP load file, refer to the CCP section of this chapter.

The file PFGRHP1 on the permanent file tape contains the NAMSTRT records that allow PTF/QTF to transfer files over a NAM/CCP or NAM/CDCNET network.

For more information about the LIDCMid file and LDT CMRDECK entry, refer to the LID/RHF Configuration Files chapter of the NOS Version 2 Analysis Handbook. For additional information about the LDT CMRDECK entry, refer to the Deadstart Decks chapter of that manual. For information about the NDL entries, refer to the Network Definition Language Reference Manual.

For information about the CCP NLFFILE, as well as sample configurations that use PTF/QTF, refer to the CCP section of this chapter. For information about the DEFINE_NP_GW command, refer to the CDCNET Configuration Guide.

Unique Parameters

Parameter	Description
SUBSYS=subsys	To install the file transfer applications to interface with RHF or NAM or both subsystems, use the SUBSYS=subsys parameter. The values you can specify are RHF, NAM, or 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 7-11 shows the binaries that are built depending on which subsystem you have specified.

Table 7-11. Binaries Built

	SUBSYS=RHF	SUBSYS=NAM	SUBSYS=BOTH
PTF	MFLINK FTF0100 FTF0200	MFLINK FTF0300 FTF0400	MFLINK FTF0500 FTF0600 FTF0700
PTFS	PTFS PFS0100 PFS0200	PTFSN PFS0300 PFS0400	PTFS 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 chapter.

Parameter Description

ACNMAXC The maximum number of connections that the 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. The following is an example of a parameter change:

```
#ACNMAXC  #DEF#  2      #ACNMAXC  #2#;
```

MAXRTRY The number of retries that an application attempts to successfully complete a file transfer if errors other than temporary connection rejects occur. 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#  20D   #MAXRTRY  #20#;
```

TIMEOUT The time in seconds (assuming a job scheduler cycle of 1 second) in which a response must be received from the remote application before the connection is broken. Values may range from 1 through 1800. 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#;
```

QTF Initiation Procedure Parameters

The QTF initiation procedure has two parameters. For a full description of these parameters, refer to the NOS Version 2 Analysis Handbook.

QTF Configuration Directives

Refer to the NOS Version 2 Analysis Handbook for information about modifying QTF configuration directives.

SORT5 - Sort/Merge Version 5

This section describes the installation procedure messages for Sort/Merge Version 5.

Installation Procedure Messages

The DECKOPL installation procedure for SORT5 contains an error. The following message appears in the dayfile:

```
COPYL DID NOT FIND -- REL      /S$ISQRT
```

This condition is non-fatal and does not affect the generated binaries. The frequency of occurrence is relative to this product as released; any local code may change the frequency.

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 section describes these installation options for TAF:

- Unique Parameters
- Task Library
- TAF Procedure File
- TAF Validation Requirements
- Installation Parameters

Unique Parameters

Parameter	Description
DEBUG	To get the trace feature, specify the keyword DEBUG on the call to the TAF installation procedure.
NOLOG	To not assemble the TAFLOG binary (which requires COBOL5), specify the keyword NOLOG on the call to the TAF installation procedure.
TASKLB	To create a task library (refer to Task Library later in this section), specify the keyword TASKLB on the call to the TAF installation procedure.

Task Library

Before running the TAF installation procedure, create a task library permanent file containing the following required tasks:

Task	Description
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.
ITASK	Initial task.
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.
RCTASK	Task that recovers CDCS transactions.
RTASK	Task to recover terminals. RTASK may be on the task library permanent file or on database libraries.
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 chapter, for information about the subsystem startup file and subsystem initiation.

The user name required by TAF (KB100DC) is created automatically by SYSGEN if no validation file exists.

TAF Validation Requirements

If you have no validation file, SYSGEN creates the necessary user name and password 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 TAF file. To change the user name or user index, reassemble TAF changing the USNM and TRUI micros. These micros are set in deck COMKIPR. (File ZZSYSGU should also be updated to reflect the new user name and password. Refer to SYSGEN Validations in chapter 8 for more information.)

Installation Parameters

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 and family under which TAF runs.

Parameter	Default	Description
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.
FMLY	A null micro	Micro whose string specifies the family under which DMREC will locate TAF's xxJ files. FMLY must be set only if databases are defined on families other than the family that contains the xxJ files. That is, if all databases and xxJ files are defined on the same family, you do not need to set the FMLY parameter.
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	Description
ECSFL	0	Extended memory field length/1000 _g . ECSFL cannot be less than 0 nor greater than 400 _g .
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 _g nor greater than 376600 _g , 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	Description
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 .

Unless otherwise specified, the following parameters are defined in deck COMKIPR. These parameters specify the default time dependencies. Although these values are expressed in milliseconds, they are accurate to only 1 second.

Parameter	Default	Description
CORTL	1*1000	Frequency TAF checks to see if memory can be released to the system.
DMMTL	4	Time allowed to elapse between calls to the data managers.
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.
TSKTL	120	Task time slice in milliseconds.

The following parameters, defined in deck TAF, specify default task rollout parameters.

Parameter	Default	Description
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.

Unless otherwise specified, the following parameters are defined in deck COMKIPR. These parameters specify other default TAF installation parameters.

Parameter	Default	Description																										
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.																										
DTYM	DI	Micro whose string specifies the device type for journal files.																										
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.																										
IPTAR	1	Automatic recovery flag: <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Automatic recovery is disabled.</td> </tr> <tr> <td>1</td> <td>Automatic recovery is enabled.</td> </tr> </tbody> </table> <p>If recovery is disabled, the following requests are not honored in recovery mode.</p> <table border="1"> <thead> <tr> <th>Request</th> <th>Comments</th> </tr> </thead> <tbody> <tr> <td>CALLRTN</td> <td>Transactions can be scheduled, but input is not logged to the communication recovery file (CRF).</td> </tr> <tr> <td>RERUN</td> <td></td> </tr> <tr> <td>RGET</td> <td></td> </tr> <tr> <td>RPUT</td> <td></td> </tr> <tr> <td>RSECURE</td> <td></td> </tr> <tr> <td>SECURE</td> <td></td> </tr> <tr> <td>TINVOKE</td> <td></td> </tr> <tr> <td>TSTAT</td> <td>Except for the keywords USER and NEXT.</td> </tr> <tr> <td>WSTAT</td> <td>Except for the keywords STEP (=8 or =9) and USER.</td> </tr> </tbody> </table>	Value	Description	0	Automatic recovery is disabled.	1	Automatic recovery is enabled.	Request	Comments	CALLRTN	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.
Value	Description																											
0	Automatic recovery is disabled.																											
1	Automatic recovery is enabled.																											
Request	Comments																											
CALLRTN	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	Description
IPTST	500	Number of terminals that can access TAF. IPTST must be greater than 0 and less than 4095.
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 a 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.
NCTL	250	Maximum number of terminals in the 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.

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.

RECDF	0	Default user recovery flag:						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>User recovery is enabled.</td> </tr> <tr> <td>1</td> <td>User recovery is disabled.</td> </tr> </tbody> </table>	Value	Description	0	User recovery is enabled.	1	User recovery is disabled.
Value	Description							
0	User recovery is enabled.							
1	User recovery is disabled.							
TLDL	TLDLE*10	<p>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).</p> <p>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 by using the TT option are available to be executed. TLDL is defined in deck COMKTLD.</p>						

Parameter	Default	Description
TYPEAH	0	Enables or disables a user to type ahead in TAF.
		Value Description
	0	Type-ahead is enabled.
	1	Type-ahead is disabled.

The following parameters are used with the TAF/CRM Data Manager and are defined in deck COMKIPR.

Parameter	Default	Description
AIBFL	31	Input queue length.
AOBFL	31	Output queue length.
BMAX	8	Number of before-image recovery files. The maximum value for BMAX is 63.
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	70000B	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 the CRM Advanced Access Methods Version 2 Reference Manual).
CRMUPM	15	Number of updates allowed. Also defines the number of records in each segment of the before-image recovery files.
RMDM	1	Number of mainframes running the 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 Version 2 Applications Installation Handbook.

Parameter	Default	Description
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 parameter is defined in deck COMKNWC.

Parameter	Default	Description
MLIM	100	Maximum number of words in one SEND request before a task is rolled out pending completion of terminal output.

Unless otherwise specified, the following parameters are defined in deck COMKIPR. These parameters specify the default communication block parameters.

Parameter	Default	Description
CBDL	57	Length of the data input area in the communication blocks. This parameter is in deck COMKCBDB.
CBUL	9	Length of user area in the communication blocks. This parameter is in deck COMKCBDB.
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 $n - NLIN * NCBC - RSCMB$ 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	Description
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.
TLDMT	600	Maximum number of tasks allowed in a library. The maximum value for TLDMT is 1353.
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	Description
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 7-track tapes
		1 9-track tapes

Parameter	Default	Description												
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 _g	Number of dumps or directives. NDUMP must be less than 500 _g .												
NUMARF	1	Number of duplicate ARF copies.												
TDEN	4	Tape density for dumps; any of the following. <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>556 cpi</td> </tr> <tr> <td>2</td> <td>200 cpi</td> </tr> <tr> <td>3</td> <td>800 cpi</td> </tr> <tr> <td>4</td> <td>1600 cpi</td> </tr> <tr> <td>5</td> <td>6250 cpi</td> </tr> </tbody> </table> <p>This parameter is in deck COMKIPR.</p>	Value	Description	1	556 cpi	2	200 cpi	3	800 cpi	4	1600 cpi	5	6250 cpi
Value	Description													
1	556 cpi													
2	200 cpi													
3	800 cpi													
4	1600 cpi													
5	6250 cpi													
TDTR	200 _g +40 _g * DTTP+ TDEN	Tape format definition.												
TLOGL	100	Number of files in database.												
TTIGL	5000	Length of table that contains the transaction entries to be ignored during DMREC recovery.												
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).												

TCP/IP/FTP/TELNET

The TCP/IP/TELNET portion of the software (that is, software that executes in the DI) is automatically installed as part of a CDCNET installation.

File Transfer Protocol (FTP) consists of three NAM applications that provide file transfer capabilities between hosts on a TCP/IP network. FTP can be utilized only on a system connected to a CDCNET network that has access to a configured TCP/IP gateway.

The NAM applications are built as two separate binaries: FTP and FTPS. The FTPS binary has two entry points: FTPI (client mode server) and FTSP (server mode server). FTP is invoked by the NOS user to initiate a file transfer session. The FTPI and FTSP applications are started by NAM as part of the network startup. Subsequent copies of the FTPI and FTSP applications are started by NAM based on a user request.

This section describes the following:

- Unique Parameters
- Host Name and Address Definition for TCP/IP Network
- NAM Startup Procedure File Changes
- FTP Network Host Application Program Requirements
- Network Definition Language (NDL) Requirements

Unique Parameters

Parameter	Description
DEBUG	To add code to aid in debugging and maintenance, specify the keyword DEBUG on the call to the TCPH installation procedure.
TRACE	To create the trace and log files, specify the keyword TRACE on the call to the TCPH installation procedure.

Host Name and Address Definition for TCP/IP Network

You must create an ASCII (6/12) file called TCPHOST that contains mappings between the Internet addresses and the names and aliases for hosts on the network. The TCPHOST file must be public and can be either indirect or direct access. TCPHOST must be resident under the user name specified by the NETUN2 parameter. The released default for NETUN2 is NETADMN.

Entry format:

```
internet_address system_name [ LOCALHOST_mi ] [ [alias] ... ] # Optional Comment
```

Parameter	Description
internet_address	The Internet Protocol (IP) address of the host in dotted decimal format. Refer to the CDCNET Configuration Guide.
system_name	The name of the host associated with the IP address. It must be unique across the network. It can be any string up to 31 characters. The first character must be alphabetic. Uppercase and lowercase characters are considered equivalent. Embedded blanks are not permitted.
LOCALHOST_mi	An alias that identifies the NOS host. mi is the NOS machine id. This alias identifies the entry for the NOS host on which FTP executes. The host uses this entry to determine its IP address. If more than one NOS host shares the permanent file base where the TCPHOST file resides, a LOCALHOST_mi entry is required for each host that has access to the TCP/IP network.
alias	An alternate name for the host. Typically, it would be a shorter name. More than one alias for a given host is permitted. An alias need not be unique across the network, however, an alias must be unique in each host file, since the mapping of an alias to an IP address applies to the first match found in the file.
#	Indicates the beginning of a comment.

The following example shows a possible TCPHOST file:

```
#
# Host file for hosts S05 and S04
#

192.5.209.100   BDI_F_V           # BDI_E - FTP/VE BDI
192.5.209.101   CYBER_S0          # C930 S/N 106 SVL
192.5.209.102   S05 s05 localhost_05 # C835 s/n 102 NOS
192.5.209.103   V05 v05          # C835 s/n 102 NOS/VE
192.5.209.104   S04 s04 localhost_04 # C830 s/n 902 NOS
192.5.209.105   V04 v04          # C830 s/n 902 NOS/VE
192.5.209.120   SVLBDIS          # SVL CLSH SERVER DI
192.5.209.121   SVLBDIU          # SVL CLSH USER DI
192.5.209.122   S01 s01          # C855 s/n 125 NOS
192.5.209.123   V01 v01          # C855 s/n 125 NOS/VE
192.5.209.124   S02 s02          # C855 s/n 106 NOS
192.5.209.125   V02 v02          # C855 s/n 106 NOS/VE
192.5.209.126   M07 m07          # C875 s/n 907 NOS (ARH)
192.5.209.127   S03 s03          # C760 S/N 407
192.5.209.130   AN02             # C990 S/N 102 NOS (ARH)
192.5.209.131   AV02             # C990 S/N 102 NOS/VE (ARH)
192.5.209.53    MERCURY VAX_e vax  # VAX 11/750
192.5.209.54    COMPAQ            # COMPAQ
192.5.209.55    ICARUS SUN        # SUN
192.5.209.57    IRIS C910         # CYBER 910 / SILICON GRAPHICS
192.5.209.58    CLOVER            # CYBER 910 / SILICON GRAPHICS
```

NAM Startup Procedure File Changes

FTP applications use the same NAMI job startup procedures as does NAM, with the following additions to NAMSTRT:

- Application startup calls in the network startup records.

Call	Description
JOB(JOBFTPI,IP,SY,NS)	File Transfer Protocol Interpreter/Server (FTPI) – Normal Job
JOB(JOBFTIR,PI,DI,SY,NS)	File Transfer Protocol Interpreter/Server (FTPI) – Restart Job
JOB(JOBFTPS,ST,SY,NS)	File Transfer Protocol Server (FTPS) – Normal Job
JOB(JOBFTSR,TS,DI,SY,NS)	File Transfer Protocol Server (FTPS) – Restart Job

- Parameters for the standard NAM startup records:

Call	Description
PARAM(PISTART=YES)	Start the initial copy of the client mode FTP File Server (FTPI) when NAM comes up.
PARAM(TSSTART=YES)	Start the initial copy of the server mode FTP File Server (FTPS) when NAM comes up.

NOTE

A NO value for either of the above parameters inhibits starting of the corresponding FTP data transfer application. If FTPI is not started, FTP client mode access from the CYBER host to the FTP server on other TCP/IP hosts in the network is unavailable. If FTPS is not started, FTP server mode access from other TCP/IP hosts in the network to the CYBER host is unavailable.

FTP Network Host Application Program Requirements

Job skeleton records for the FTPI and FTPS jobs must contain a command that calls the program the job intends to run. These commands have the following format:

```
prog(parameter1, . . . , parameterN)
```

Field	Description
prog	Program call. May be FTPI or FTPS.
parameter _i	Unless specified otherwise, these are optional, order-independent parameters. They may be of the form: keyword=value

The files used by each program are listed following the description of each program call.

FTPI - File Transfer Protocol Interpreter/Server

FTPI requires the following command:

```
FTPI (NIN=nin,MC=mc,TCPHUN=tcphun)
```

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=0. The default value, which is 500, may be reset using the ZZMC PARAM statement.
NIN=nin	Network invocation number. One- to three-digit decimal number assigned by NAM when the network operation is started. This parameter is required. It is used to create trace file names.
TCPHUN=tcphun	The user name where the TCPHOST file resides. This parameter is required.

FTPI requires the following files:

File	Description
ZZZZZ1	The job record to be copied to the ZZZZZDN file by NETREL. This job record determines the disposition of the network trace file when NETREL is called on a periodic basis.
ZZZZZ2	The job record to be copied to the ZZZZZDN file by NETREL. This job record determines the disposition of the network trace file when NETREL is called as a result of an operator command.

NOTE

The default job skeleton of FTPI creates ZZZZZ1 and ZZZZZ2 from the input records of the FTPI job.

FTPS - File Transfer Protocol Server

FTPS requires the following command:

```
FTPS (NIN=nin,MC=mc,TCPHUN=tcphun)
```

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=0. The default value, which is 500, may be reset using the ZZMC PARAM statement.

Parameter	Description
NIN=nin	Network invocation number. One- to three-digit decimal number assigned by NAM when the network operation is started. This parameter is required. It is used to create trace file names.
TCPHUN=tcphun	The username where the TCPHOST file resides. This parameter is required.

FTPS requires the following files:

File	Description
ZZZZZN1 and ZZZZZN2	Described under FTPI, earlier in this section.

NOTE

The default job skeleton of FTPS creates ZZZZZN1 and ZZZZZN2 from the input records of the FTPS job.

Network Definition Language (NDL) Requirements

To use FTP with CDCNET, you must modify the Local Configuration File (LCF) to include application definition (APPL) statements and OUTCALL statements. The APPL statements define the NAM applications required to support FTP, and the OUTCALL statements define the path used by the FTP applications to access the CDCNET TCP/IP gateway. The released NDL file, NDLDATA, on the network administration user name contains the required statements. They are as follows:

```

LCFFILE      :LFILE.
FTP          :APPL, MXCOPYS = 15.
FTPI        :APPL, MXCOPYS = 15, PRIV.
FTPS        :APPL, MXCOPYS = 15, PRIV.
,
,
*** OUTCALL BLOCK TO ACCESS TCP/IP GATEWAY
*
* REPLACE THE TEXT SURROUNDED BY POUND SIGNS (FOR EXAMPLE, #DN#)
* BELOW WITH THE MACHINE ID FOR THE NOS HOST, THE NODE NUMBERS
* FOR THE MDI USED TO ACCESS CDCNET, AND THE CYBER HOST MODEL AND
* SERIAL NUMBER. REMOVE THE ASTERISK FROM COLUMN ONE TO ACTIVATE
* THE OUTCALL FOR FTP/NOS.
*
* IF MULTIPLE CYBER HOSTS ARE SHARING THE PERMANENT FILES WHERE
* THE TCPHOST FILE RESIDES, AN OUTCALL ENTRY IS REQUIRED FOR
* EACH HOST THAT HAS ACCESS TO THE TCP/IP NETWORK.
,
,
,
* OUTCALL      NAME1 = TCPIPGW, NAME2 = H#MI#, SNODE = #SN#, DNODE = #DN#,
*              NETOSD = DDV, ABL = 7, DBL = 7, DBZ = 2000, UBL = 7,
*              UBZ = 18, SERVICE = GW_TCPIP_#MMM#_#SSS#.
*
END.
```


Parameter	Description
#DN#	The destination node is the decimal value of the NT parameter on the MDI's EQPDECK entry. Refer to the example below.
#MI#	The NOS machine id is the two alphanumeric characters from the MID CMRDECK entry.
#MMM#	The CYBER model number (leading zeroes removed). For example, if your mainframe is a 180-810, 810 would be specified for this parameter.
#SN#	The source node is the decimal value of the ND parameter on the MDI's EQPDECK entry. It is channel-connected to the NOS host #MI# and has access to the CDCNET containing the TCP/IP gateway. Refer to the example below.
#SSS#	The CYBER serial number (leading zeroes removed). For example, if your CYBER serial number is 002, only the 2 would be specified for this parameter.

This example shows the relationship between the EQPDECK entry and the OUTCALL parameters. If the EQPDECK entry is as follows:

```
EQ033=ND, ST=ON, EQ=00, PI=1, CH=27, ND=45, NT=65.
```

The #SN# parameter would be 37, and the #DN# parameter would be 53.

For more information, refer to the Network Definition Language Reference Manual and the CDCNET Configuration Guide.

TERMLIB - Terminal Library

TERMLIB creates the permanent files TERMLIB and TDUFILE and places terminal capsules (termcaps) which the site has chosen to be resident termcaps into SFLIB and QSFLIB.

This section describes the unique parameters in TERMLIB and gives an example of how to call this procedure.

Unique Parameters

Parameter	Description
-----------	-------------

TERMCAP	This parameter allows you to specify which terminal capsules will be resident. The default is no terminal capsules are declared resident.
---------	---

NOTE

To make a terminal capsule resident, you must modify the common deck COMCGTO to match the capsules specified with the TERMCAP parameter.

Example:

The following installation procedure call to TERMLIB changes the default resident terminal capsule from none to Z721, Z722, and ZVT100:

```
BEGIN, TERMLIB, INSTALL, TERMCAP=$Z721, Z722, ZVT100$
```

TEXT and TEXTIO - Product TEXTS and Product TEXTS I/O

This section describes these installation options for TEXT and TEXTIO:

- Default IPARAMS
- Unique Parameters
- Installation Parameters
- MFT Parameter Details
- Installation Procedure Messages

Default IPARAMS

General installation parameters related to the common products are defined within the common deck IPARAMS, which is included in product TEXT.

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 as well as the micros set by the unique parameters, run a job similar to the following:

```

job command.
USER, insun, password.
BEGIN, PRDIN, INSTALL, PRDNAME=TEXT, DISK=0.
UPDATE, Q.
COMPASS, A, I, S=0.
GTR, INSTALL, Z1.PROC/TEXT
COPYSBF, Z1, OUTPUT.
--eor--
*COMPILE IPTEXT
--eoi--

```

The CSET, ECS, and MFT unique parameters also set micros and eliminate the need to create a USER file for several of the micros. Refer to the TEXT installation procedure in DECKOPL for more information.

Unique Parameters

Parameter	Description
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:

Value	Description
A63	Selects the ASCII graphic 63 character set.
A64	Selects the ASCII graphic 64 character set.
C63	Selects the CDC graphic 63 character set.
C64	Selects the CDC graphic 64 character set.

This parameter sets the IP.CSET symbol; the following reference IP.CSET:

AAM 2	COBOL 5	FCL 2	Query Update 3
APL 2	COMPASS 3	FCL 5	Update 1
BASIC 3	FCL 1	FORTTRAN 5	8-Bit Subroutines 1

Parameter	Description
-----------	-------------

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. Note that extended memory is available only if UEC is defined by the EQPDECK XM entry during deadstart. (Refer to the NOS Version 2 Analysis Handbook for information about EQPDECK entries.) 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, 76, 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 870, 875, 960, 990, 994, or 995. This parameter sets the MODEL, HF.LIST, and IP.CMU symbols; most common products reference the MODEL micro. For more information, refer to MFT Parameter Details, later in this section.

NOTE

Do not add user code to set the symbols HF.LIST, IP.CMU, IP.CSET, IP.MECS, and MODEL, since they are set by the above parameters.

Installation Parameters

There is only one installation-changeable symbol in IPARAMS.

Parameter	Default	Description
OS.ID	NOS x.x.x	System identification micro for displaying the operating system name and version number in generated program binaries (for example NOS 2.5.2). Most common products reference the OS.ID micro.

MFT Parameter Details

The HF.LIST, IP.CMU, and MODEL micros are set by the MFT parameter. To determine the relationship between MFT and these micros, consult a listing of the TEXT installation procedure in DECKOPL. 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.

Parameter	Description								
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. COBOL 5 references IP.CMU.								
MODEL	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, 76, 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 870, 875, 960, 990, 994, or 995. This parameter sets the MODEL symbol. Most common products reference the MODEL micro.								
HF.LIST	Micro whose value specifies the presence of certain hardware features in the configuration on which the products are being used. HF.LIST is set in addition to the MODEL micro, since use of various hardware features by the products is conditional on HF.LIST. HF.LIST controls the following: <table border="1"> <thead> <tr> <th>Entry</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>C</td> <td>Compare/move unit (CMU) hardware is present.</td> </tr> <tr> <td>L</td> <td>For models 76 and 176, LCME is present. For models 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 870, 875, 960, 990, 994, and 995, 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.</td> </tr> <tr> <td>Sn</td> <td>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 one of the following. 7 74 and 6600 10 76, 175, 176, 740, 750, 760, 865, and 875 48 990, 994, 995</td> </tr> </tbody> </table>	Entry	Description	C	Compare/move unit (CMU) hardware is present.	L	For models 76 and 176, LCME is present. For models 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 870, 875, 960, 990, 994, and 995, 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 one of the following. 7 74 and 6600 10 76, 175, 176, 740, 750, 760, 865, and 875 48 990, 994, 995
Entry	Description								
C	Compare/move unit (CMU) hardware is present.								
L	For models 76 and 176, LCME is present. For models 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 870, 875, 960, 990, 994, and 995, 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 one of the following. 7 74 and 6600 10 76, 175, 176, 740, 750, 760, 865, and 875 48 990, 994, 995								

Parameter Description

HF.LIST (continued)

Entry Description

ES Central processor exits a block copy sequentially into the next parcel of the current instruction word if no errors occur. Model 76 only.

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, 860, 870, and 960; serial type CPU, etc.

74

6600, 6700, and 74

76

76

175

175, 740, 750, and 760

176

176

740

740, 175, 750, and 760

750

750, 175, 740, and 760

760

760, 175, 740, and 750

865

865, 875

875

875, 865

990

990, 994, 995

994

994, 990, 995

995

995, 990, 994

Parameter Description

HF.LIST	(continued)
	Entry Description
	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 (models 76 and 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, 870, 875, 960, 990, 994, or 995.

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
76	P76,S10,L,ES,PSD
171	PS
172	C,PS
173	C,PS
174	C,PS
175	P175,S10
176	P176,S10,L,PSD
720	C,PS
730	C,PS
740	P740,S10
750	P750,S10
760	P760,S10
810	C,PS,CRW,L
815	C,PS,CRW,L
825	C,PS,CRW,L
830	C,PS,CRW,L
835	PS,CRW,L
840	PS,CRW,L
845	PS,CRW,L
850	PS,CRW,L
855	PS,CRW,L
860	PS,CRW,L
865	P865,S10,CRW,L
870	PS,CRW,L
875	P875,S10,CRW,L
960	PS,CRW,L
990	P990,S48,CRW,L
994	P994,S48,CRW,L

MODEL	HF.LIST
Micro Value	Default String
995	P995,S48,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.

Installation Procedure Messages

The DECKOPL installation procedure for TEXT contains an update error. The following message appears in the load map:

```
0*** YANK, SELYANK, OR YANKDECK IDENT 1EP NOT KNOWN ***
```

The following message appears in the dayfile:

```
1 NON-FATAL ERRORS.
```

This condition is non-fatal and does not affect the generated binaries. The frequency of occurrence is relative to this product as released; any local code may change the frequency.

UPDATE (Common Memory Manager Version 1, CYBER Common Utilities, 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	Description
AUDITKEY	Enabled	Allows audit functions.
CHAR64	Enabled	Declares 64-character set Update program library output.
DECLKEY	Enabled	Enables DECLARE directive.
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.
EDITKEY	Enabled	Allows merge and edit functions.
EXTOVLP	Enabled	Enables detection of four types of overlap involving two or more cards in a correction set.
OLDPLKEY	Enabled	Enables Update to read both old-style and new-style old program libraries.
PMODKEY	Enabled	Enables PULLMOD statement and G option.

The 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	Description						
DEFVER	0	Defines which of two versions of CMM is to be used by default.						
		<table border="1"> <thead> <tr> <th>Value</th> <th>Parameter</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>A version without error checking (FAST) is used.</td> </tr> <tr> <td>1</td> <td>An error checking version (SAFE) is used.</td> </tr> </tbody> </table>	Value	Parameter	0	A version without error checking (FAST) is used.	1	An error checking version (SAFE) is used.
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.						

Introduction	8-1
Calling SYSGEN	8-1
Loading Permanent Files	8-2
Loading the RECLAIM Database	8-2
Loading the Files	8-3
SYSGEN(xxxx) Functions	8-3
SYSGEN(group) Functions	8-4
SYSGEN(FILE)	8-4
SYSGEN(FULL)	8-4
SYSGEN(RECLAIM)	8-5
SYSGEN(SOURCE)	8-5
SYSGEN Functions	8-5
SYSGEN(COPYSYS)	8-5
SYSGEN(COPYTAP)	8-6
SYSGEN(DST)	8-7
SYSGEN(MOVE)	8-7
SYSGEN(SWAP)	8-8
SYSGEN(UPGRADE)	8-9
SYSGEN Maintenance	8-9
SYSGEN Validations	8-10

Introduction

This chapter describes the SYSGEN installation command in the following sections:

- Calling SYSGEN
- Loading Permanent Files
- SYSGEN Functions
- SYSGEN Maintenance
- SYSGEN Validations

Calling SYSGEN

SYSGEN functions may be executed from DSD, DIS, an interactive terminal, or a batch job. The method used for calling SYSGEN determines how SYSGEN will load the files.

- To call SYSGEN from DSD, use this command format:

```
X.SYSGEN(function,params)
```

SYSGEN loads any permanent files directly to the user name where the files are supposed to reside. To do this, you must have correct validations. Consult SYSGEN Validations later in this chapter for complete information.

- To call SYSGEN from DIS, use this command format:

```
USER(username,password)      -OR-      SUI(userindex)  
SYSGEN(function,params)      SYSGEN(function,params)
```

SYSGEN loads any permanent files directly to the user name/index you have specified, unless the user name/index is UN=SYSTEMX (UI=377777B). When SYSGEN is initiated from user name SYSTEMX, USER commands are executed and files are installed in the same manner as when SYSGEN is called from DSD. If SYSGEN is not started from user name SYSTEMX, USER commands are not executed. This will install the same as when SYSGEN is called from an interactive or batch job.

- To call SYSGEN from an interactive terminal or from within a batch job, use this command format:

```
SYSGEN(function,params)
```

SYSGEN loads any permanent files directly to the user name under which the interactive or batch job is running.

Loading Permanent Files

This section documents the SYSGEN functions that load one or more permanent files from the permanent file tapes:

- Loading the RECLAIM database
- Loading the files

If you have not deadstarted your new system, you will need to load the RECLAIM database.

Loading the RECLAIM Database

SYSGEN(INIT) is used to put the database on disk for upgrade orders and SYSGEN(UPGRADE) is used for component orders. Both functions may be executed from either an interactive terminal or from DIS at the system console.

- To install an upgrade order:
 1. Mount the released deadstart tape on an available unit.
 2. To load files from an interactive terminal, log in to the installation user name to set up the RECLAIM database. To load files from DIS at the system console, first do a USER or SUI command to the installation user name.
 3. Enter the following commands:

```
REQUEST, TAPE, VSN=vs n, D=density, LB=KU, F=I, PO=R.  
COPYEI, TAPE, SYSTEM, V.  
UNLOAD, TAPE.  
BEGIN, SYSGEN, SYSTEM, INIT, UPGRADE.
```

Replace vsn with the VSN of the released deadstart tape (contained in the media number field of the external tape label). Replace density with the tape density of the tape (HY, HD, PE, or GE).

The BEGIN call invokes SYSGEN(INIT), which loads the RECLAIM database from the deadstart tape and stores it on the user name under which the commands were executed. This also places a copy of the SYSGEN procedure on this user name and the user library PFGLIB in file PRODUCT, which allows the SYSGEN procedures from the new release tape to be used in multiple terminal sessions and from multiple user names. Binaries of NDLP, NETFM, and other utilities are placed in GLOBLIB to facilitate easier upgrades. Any RECLDB file that you may have had before the BEGIN command was executed will be replaced.

Once INIT has completed, you may immediately execute the SYSGEN functions described in Loading the Files later in this section. However, if you wish to use SYSGEN during multiple terminal sessions or from user names other than the installation user name, you must do the following before executing the SYSGEN command:

```
GET, SYSGEN/UN=insun.
```

This gets the newly released version of SYSGEN, which in turn gets the newly released PFGLIB from file PRODUCT. This is also on the installation user name. (PFGLIB contains all SYSGEN procedures.)

NOTE

Do not initiate SYSGEN from DSD using X.SYSGEN function calls until you have deadstarted your new system. If you initiate SYSGEN before deadstarting the new system, you will be using an old and possibly incompatible version of SYSGEN.

- To install a component order:
 1. Mount the released permanent file tape on an available tape unit.
 2. To load files from an interactive terminal, log in to the installation user name to set up the RECLAIM database. To load files from DIS at the system console, first do a USER command to the installation user name.

NOTE

Execute a USER command before executing SYSGEN(UPGRADE) from the system console. SYSGEN(UPGRADE) will not work correctly if a SUI command is used.

3. Enter the following command:

```
SYSGEN,UPGRADE,0,0,vsn,density.
```

Replace vsn with the VSN of the first permanent file tape; replace density with the density of the tape.

This command updates your RECLAIM database (or creates one) with information about files on the permanent file tapes. Once you set up the RECLAIM database on your installation user name, you may load files from it.

Loading the Files

When loading permanent files, you may install files directly to the appropriate user name, or you may load files to a different user name for examination or modification, then move them to the correct user name later. For example, you may load file NAMSTRT directly to the network operations user name or load it to the installation user name for modification, then move it to the network operations user name later. This is determined by how SYSGEN is called. Refer to Calling SYSGEN, earlier in this chapter.

SYSGEN(XXXX) Functions

Use the SYSGEN(XXXX) function to load each PFGXXXX file from the permanent file tape and install the contents of the file. Replace XXXX with the matching XXXX of the PFGXXXX permanent file group name (PFGXXXX names are listed in appendix B). The following example loads and installs the files in PFGDECK:

```
SYSGEN (DECK)
```

SYSGEN(group) Functions

Use SYSGEN(group) functions to load a group of individual PFGxxxx files all at one time. SYSGEN has the following groups:

Group	Description
CDCNET	Installs components of CDCNET. When executed interactively, this group does not install all the files; additional steps are required. Refer to the CDCNET section of chapter 7.
LIBRARY	Installs all required files for user name LIBRARY.
NAMSTRT	Installs all products that update the NAMSTRT file.
OTHER	Miscellaneous products.
SITE	Installs standard installation site tools.
SUBSYS	Installs all subsystem startup files (excludes NAMSTRT).

Consult the SYSGEN procedures for exact details on which files belong with which groups. Note that there is some overlap, for example, PFGFSE1 is used as part of SITE, SUBSYS, and LIBRARY.

SYSGEN(FILE)

Use this function to load permanent files from the permanent file tapes when installing an upgrade or component order. All files are loaded except those needed to build products from source and those that were deleted from the database (as described in chapters 3 and 4). Validation files are not created or changed. SYSGEN(FILE) calls the following group functions (explained earlier in this section): SITE, NAMSTRT, SUBSYS, LIBRARY, OTHER and CDCNET.

SYSGEN(FILE) has one parameter called vsn. Replace vsn with the VSN listed in the media number field on the external tape label of the first permanent file tape. SYSGEN(FILE) may be executed from DSD or from DIS under username SYSTEMX. Refer to Calling SYSGEN, earlier in this chapter.

SYSGEN(FULL)

Use this function to load permanent files from the permanent file tapes during a first-time installation. Validation files are created and ZZSYSGU is placed on user name SYSTEMX (refer to SYSGEN Validations later in this chapter for more information). Next, the RECLAIM database is loaded to user name INSTALL and the SYSGEN(FILE) function is called.

SYSGEN(FULL) has no parameters. SYSGEN(FULL) should be executed from DSD. Refer to Calling SYSGEN, earlier in this chapter.

SYSGEN(RECLAIM)

Use this function to load one to five permanent files from the permanent file tapes. These files are left local to your job. The RECLAIM database must have been previously loaded to the installation user name for this function to work properly. (This is done automatically during the SYSGEN(FULL) and SYSGEN(SOURCE) functions.)

SYSGEN(RECLAIM) has five optional parameters. Replace them with the one to five file names that you want loaded. SYSGEN(RECLAIM) should be executed only from DIS, an interactive terminal, or a batch job. Refer to Calling SYSGEN, earlier in this chapter.

SYSGEN(SOURCE)

Use this function to load the permanent files needed to perform a customized installation. If the RECLAIM database does not exist on the installation user name, it is loaded there. Other files loaded at this time include: DECKOPL, INSTALL, COMMOD, GDIR, CODEPL, DBUBIN, MISCPL, PSRRPT, USERBPS, and RECLAIM.

SYSGEN(SOURCE) has one keyword parameter called CCP. Include this keyword if you want to start building CCP at the CCPVAR step (refer to the CCP section of chapter 7). SYSGEN(SOURCE) may be executed following any of the formats described in Calling SYSGEN, earlier in this chapter.

SYSGEN Functions

This section documents the SYSGEN functions not related to the loading of permanent files:

- SYSGEN(COPYSYS) copies the running system to disk or tape.
- SYSGEN(COPYTAP) makes a copy of the release tapes.
- SYSGEN(DST) creates a new deadstart tape.
- SYSGEN(MOVE) moves permanent files.
- SYSGEN(SWAP) adds debug/trace/63-character set binaries to the deadstart tape.
- SYSGEN(UPGRADE) installs component orders.

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 CIP tape. To use this function, you need two tape drives. Use this format to call the function:

```
X.SYSGEN(COPYTAP,aaaaa,density,numpf,dst,cip)
```

Parameter Description

Parameter	Description
aaaaa	Specifies the first 5 characters of the VSN of the permanent file tape.
cip	Specifies whether to copy the CIP tape. Enter YES or NO.
density	Specifies the density of the release tapes.
dst	Specifies whether to copy the deadstart tape. Enter YES or NO.
numpf	Specifies the number of permanent file tapes to copy. Specify 0 if you do not want to copy any of the tapes.

If you specify that permanent file tapes should be copied, the function writes new labels (identical to those of the original tapes) on numpf 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, if specified.

When you use the SYSGEN(COPYTAP) command to make copies of the permanent file tapes, make sure that you copy the files to a tape that is the exact same length as the original release tape. However, even when you use same-length tapes, the position of a file can shift because tapes can vary in the number of usable feet.

If for some reason the position of a file shifts on the tape, you may have problems extracting the files from the tape. When RECLAIM cannot find a file on the tape, it issues the following message:

```
DUMP FILE MALFUNCTION - FILE NAME MISMATCH
```

You can rebuild the database and correct the problem by executing these commands from the installation user name where the RECLAIM database is stored:

```
ATTACH (RECLDB/M=W)
EVICT (RECLDB)
RETURN (RECLDB)
RECLAIM(Z,L=LIST)/LIST,UN=NS2psrin,TN=vsn,D=density,xx.
```

Replace psrin with the 3-digit PSR level of the NOS release; replace vsn with the volume serial number of the first permanent file tape; replace density with the tape density; replace xx with MT for a 7-track tape or NT for a 9-track tape. RECLAIM reads all the tapes and rebuilds the database.

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 chapter 9). Use this format to call the function:

```
SYSGEN(DST,old,lgo,new,input,density)
```

Parameter Description

Parameter	Description
density	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.
input	Name of a file containing directives for the LIBEDIT that SYSGEN performs. Specify 0 (zero) if you have no LIBEDIT directives.
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 used.
new	Name of file to receive the new deadstart file. If no file is assigned and the density parameter is specified, a tape request for VSN=NDT is made. This file can be a preassigned magnetic tape.
old	Name of the base deadstart file. This deadstart file should be local to your job or you should supply the density parameter so that SYSGEN can request a deadstart tape to use as the base. If a tape is requested, the VSN is ODT. This file can be a pre-assigned magnetic tape file.

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 later in this chapter. You must execute SYSGEN(MOVE) from the system console. Use this format to call the function:

```
X.SYSGEN(MOVE,pfn,un,ct,ac,m)
```

Parameter	Description
ac	Specifies the alternate user CATLIST option. Enter Y to allow alternate users to CATLIST the file. Enter N to prevent alternate users from using CATLIST to list the file. The default is N.
ct	Specifies the permanent file category. Enter PU for public, PR for private, or S for semiprivate. The default is PR.
m	Assigns read permission to the installation user name. Enter PERMIT to assign read permission. The default is to not assign read permission. Use PERMIT only when ct is set to PR.
pfm	Name of the permanent file to be moved. The file must reside on the installation user name. This parameter is required.
un	User name to receive files. Valid user names and passwords are listed in table 8-1. This parameter is required.

SYSGEN(SWAP)

This function creates a new deadstart tape with the debug/trace versions of NAM and RBF or a 63-character set version of NIP5870. Use this format to call the function:

```
X.SYSGEN(SWAP,density,p1,p2,p3)
```

Parameter	Description
density	Specifies the density of the new deadstart tape.
p _i	Specify the products to be swapped. You can specify RBF, NAM, or NIP. Any number or combination of these three can be used.

NOTE

Be sure to keep a copy of the deadstart tape that contains the original binaries.

SYSGEN(UPGRADE)

This function is used during a component order. If you are executing this function from the system console, you must do it from DIS. Use a USER command (not a SUI command) to go to the installation user name. SYSGEN(UPGRADE) performs two tasks: it updates the RECLAIM database to include information about the permanent file tapes and merges the component order binaries (if any) with a deadstart file. (It is possible to have SYSGEN(UPGRADE) only update the RECLAIM database and not have it merge any binaries.) The basic format for SYSGEN(UPGRADE) is:

```
SYSGEN(UPGRADE,old,new,vsn,density1,density2)
```

Parameter	Description
density ₁	Tape density of the permanent file tapes of the component order.
density ₂	Tape density of both the old and new deadstart tapes. This parameter is optional.
new	Name of file to receive the merged binaries.
old	Name of file containing the binaries of the current system.
vsn	Volume serial number of the first permanent file tape of the component order.

Depending on how the parameters are specified, old and new can be any combination of disk or tape files. If old is local and density₂ is specified, the density₂ parameter only affects file new. If old is local and density₂ is not specified, new is a disk file. Old and new can also be pre-assigned tapes. If old is not local and density₂ is specified, an old deadstart tape is requested with a VSN of ODT and the new deadstart tape is requested with a VSN of NDT. If old and new are both 0, only the RECLAIM database is updated; no binaries are merged.

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,insun,netopun,netadun.
```

Replace insun with your installation user name, netopun with your network operations user name, and netadun with your network administration user name. The defaults for these user names are INSTALL, NETOPS, and NETADMN respectively.

The SYSGEN procedures consist of a NOS procedure SYSGEN and a user library PFGLIB on the deadstart file. The BEGIN command, shown previously, adds SYSGEN and PFGLIB 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.)

If you need to modify SYSGEN (other than the user name parameters), you should append your modifications to file COMMOD. Then, run the SETUP procedure, using MOD=COMMOD, to create a new INSTALL procedure file.

SYSGEN Validations

To install permanent files, SYSGEN must know the batch passwords of the user names listed in table 8-1. (Interactive passwords are not used by SYSGEN.) SYSGEN uses a procedure file named ZZSYSGU (indirect access file on user name SYSTEMX) that contains USER commands with passwords that match those stored in the system validation files. Initially, the passwords in ZZSYSGU match those listed in table 8-1. However, to maintain security, these passwords may be changed as often as needed without requiring any modifications to SYSGEN.

If the VALIDUS and VALINDS files exist on your system prior to installation, you must load ZZSYSGU and either modify it to match your validations or modify your validation files to match those needed by SYSGEN. You can do so from the system console. Use one of the following commands. Before you begin, you must deadstart your new system. In addition, have the permanent file tapes available for mounting.

- X.SYSGEN(LOADUSE). This command loads file ZZSYSGU as a private indirect access file on user name SYSTEMX. Use an available editor to modify the contents of the file to match your system's validations.
- X.SYSGEN(MODVAL,ADD). This command loads file ZZSYSGU and modifies the files VALIDUS and VALINDS to contain the user names and passwords listed in table 8-1. User names that you do not have are created. Passwords for existing SYSGEN user names are changed to match those in the table. Any user name not needed by SYSGEN is not changed. Note that this function sets the user index of user name KB100DC to the released default value of 16B. No other created user names are assigned a specific user index.

If SYSGEN(FULL) created VALIDUS, VALINDS, and ZZSYSGU, you should alter the passwords upon completion of installation for security reasons. You can do so by first using the MODVAL command to change the system validation files (refer to the NOS Version 2 Administration Handbook). Next, use an available editor to modify file ZZSYSGU to match the changes you made to the validation files.

Certain products are released with default user names and passwords. If these user names or passwords are changed, the products may need to be rebuilt. Refer to chapter 7 for additional information. Products that could be affected are APL2, MAP, NSS, and TAF.

If you wish to change the default Control Data user names to ones that are site-defined, modify file ZZSYSGU as follows.

For example, to change user name INSTALL to user name INS700, alter the following line in file ZZSYSGU from

```
.IF, ($P1$ .EQ. $INSTALL$) .USER(INSTALL,INSTALL)
```

to

```
.IF, ($P1$ .EQ. $INSTALL$) .USER(INS700,password)
```

Only the USER portion should be changed. When SYSGEN finds a file that should be put on a specific user name, it keys off the P1 parameter. The P1 parameters must be the Control Data default values. When SYSGEN finds the matching value (in this example, INSTALL), it executes its USER statement. In this example, SYSGEN would put the file on user name INS700.

Table 8-1. SYSGEN User Names and Passwords

User Name	Batch Password	Products Affected
APL0	APL0	APL2
APL1	APL1	APL2
CDCCE	CDCCE	CML, MAP
CDCCE2	CDCCE2	MAP
INSTALL ¹	INSTALL	CCP, CDCNET, CROSS, Dual State, FSE, MAP, NAM5, NOS
KB100DC	TAFPASS	TAF
LIBRARY	LIBRARY	DCL, FSE, NOS, XEDIT
MANUALS	MANUALS	NOS and NOS Online Manuals
NETADMN ¹	NETADMN	CCP, CDCNET, NAM5
NETOPS ¹	NETOPsx	CDCNET, ITF, NAM5, PSU, PTF/QTF, RBF5, TCP/IP/FTP/TELNET
NVE	NVEX	Dual State
SSPOT	STATION	NSS
SUBFAM0	SUBFAM0	MSE
SYSTEMX	SYSTEMX	CDCS2, Dual State, FSE, IAF, MAP, MCS, MSE, NAM5, NOS, NSS, RHF, TAF

1. These user names refer to the Control Data default values for the installation user name, network operations user name, and the network administration user name. The default values are INSTALL, NETOPS, and NETADMN, respectively. If you have changed these values, substitute your site names for the Control Data default values.

Installation Commands, Parameters, and Procedures

9

Introduction	9-1
COMMOD File Parameters	9-1
DECKLIS Procedure	9-9
Disk Installations	9-10
Using PFLOAD	9-10
Disk Installation with Auxiliary Packs	9-11
GENDST Procedure	9-14
MISCGET Procedure	9-15
REPORT Procedure	9-15
RESETP Procedure	9-16
SETUP Procedure	9-17

Introduction

This chapter describes the following commands, parameters, and procedures used during installation:

- COMMOD File Parameters
- DECKLIS Procedure
- Disk Installations
- GENDST Procedure
- MISCGET Procedure
- REPORT Procedure
- RESETP Procedure
- SETUP Procedure

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. Refer to SYSGEN(SOURCE) in chapter 8.

NOTE

The COMMOD file parameters you must change to perform a disk installation are described later in this chapter.

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 parameter 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 list describes the parameters in file COMMOD.

Parameter	Description										
CCPLIST	Called from common deck COMCCPL. Used by CCP/CROSS installation procedures only (refer to chapter 7).										
D=option	Called from common deck COMDEN. Specifies the tape density. The default value is PE.										
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>HY</td> <td>800 cpi (7-track).</td> </tr> <tr> <td>HD</td> <td>800 cpi (9-track).</td> </tr> <tr> <td>PE</td> <td>1600 cpi (9-track).</td> </tr> <tr> <td>GE</td> <td>6250 cpi (9-track).</td> </tr> </tbody> </table>	Value	Description	HY	800 cpi (7-track).	HD	800 cpi (9-track).	PE	1600 cpi (9-track).	GE	6250 cpi (9-track).
Value	Description										
HY	800 cpi (7-track).										
HD	800 cpi (9-track).										
PE	1600 cpi (9-track).										
GE	6250 cpi (9-track).										
	<p>NOTE</p> <hr/> <p>This parameter only affects tape requests for deadstart tape and density for output PLs when DISKINS=NO and OUTPRD=YES.</p> <hr/>										
DISKINS	Called from common deck COMDISK. Controls the type of installation. The default is DISKINS=NO.										
	<table border="1" style="width: 100%;"> <thead> <tr> <th style="text-align: left;">Value</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>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.</td> </tr> <tr> <td>YES</td> <td>Disk installation; if you want to use auxiliary disk packs, additional parameters relating to disk pack name and type must be changed from the defaults. Refer to Disk Installations later in this chapter.</td> </tr> </tbody> </table>	Value	Description	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 defaults. Refer to Disk Installations later in this chapter.				
Value	Description										
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 defaults. Refer to Disk Installations later in this chapter.										

Parameter	Description
IA	<p>Called from common deck COMIA. Specifies how an installation procedure is submitted for execution. The default is IA=NO. If you do not specify the keyword IA, the installation procedure submits a batch job for execution unless the installation procedure is already executing as a batch job.</p> <p>If the job origin type is not batch, you can specify the keyword IA to cause immediate execution of an installation procedure. Specifying IA also causes the following:</p> <ul style="list-style-type: none"> • If the job origin type is interactive, the file OUTPUT is assigned to mass storage and at job completion or abnormal termination is renamed IAOUT. <p>If you want to assign the output back to your terminal rather than mass storage, enter this command:</p> <pre style="margin-left: 40px;">ASSIGN, TT, OUTPUT.</pre> <ul style="list-style-type: none"> • If the job is running from DIS, the installation procedure runs from DIS and does not submit a batch job. • The installation 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 installation procedures and to have GLOBAL remain in effect after the installation procedures are completed.
ICHG	<p>Called from common deck COMIUN. Specifies the valid charge and project number values for the installation user name. The default is an asterisk (*). Here is a sample value that causes the command CHARGE(1187,594N321) to be executed:</p> <pre style="margin-left: 40px;">*N=\$1187,594N321\$.</pre> <p>This parameter is significant only if IUCHG=YES.</p>
IFAMILY	<p>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.</p>

Parameter	Description										
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.										
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>You must create a USERCHG file (containing appropriate USER, CHARGE, RESOURC, and PACKNAM commands) for the installation procedures. The file can be local, direct, or indirect.</td> </tr> <tr> <td>YES</td> <td>The parameter values specified by IUN, ICHG, IFAMILY, IPW, RESOUR, and PCKNAM are used to generate a USERCHG file.</td> </tr> </tbody> </table>	Value	Description	NO	You must create a USERCHG file (containing appropriate USER, CHARGE, RESOURC, and PACKNAM commands) for the installation procedures. The file can be local, direct, or indirect.	YES	The parameter values specified by IUN, ICHG, IFAMILY, IPW, RESOUR, and PCKNAM are used to generate a USERCHG file.				
Value	Description										
NO	You must create a USERCHG file (containing appropriate USER, CHARGE, RESOURC, and PACKNAM commands) for the installation procedures. The file can be local, direct, or indirect.										
YES	The parameter values specified by IUN, ICHG, IFAMILY, IPW, RESOUR, and PCKNAM are used to generate a USERCHG file.										
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. The default is LIST=0. If filename is OUTPUT, the listing is printed with the installation procedure 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.										
MAPTYPE	Called from common deck COMLIST. Specifies the type of load map generated by the installation procedure. The default value is FULL.										
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>OFF</td> <td>No map is generated.</td> </tr> <tr> <td>PART</td> <td>Statistics and block map.</td> </tr> <tr> <td>ON</td> <td>Statistics, block map, and entry point cross references.</td> </tr> <tr> <td>FULL</td> <td>Statistics, block map, entry point cross references, and entry point map.</td> </tr> </tbody> </table>	Value	Description	OFF	No map is generated.	PART	Statistics and block map.	ON	Statistics, block map, and entry point cross references.	FULL	Statistics, block map, entry point cross references, and entry point map.
Value	Description										
OFF	No map is generated.										
PART	Statistics and block map.										
ON	Statistics, block map, and entry point cross references.										
FULL	Statistics, block map, entry point cross references, and entry point map.										
NOECS	Called from common deck COMNECS. Used by CCP/CROSS installation procedures only (refer to chapter 7).										
NOPURGE	Called from common deck COMNOPG. Used by CCP/CROSS installation procedures only (refer to chapter 7).										

Parameter	Description						
OUTPRD	Called from common deck COMDISK. Controls the production of output program libraries in installation procedures. The default is NO.						
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>Output program libraries are not generated or written.</td> </tr> <tr> <td>YES</td> <td>The installation procedures write output program libraries. This does not apply to MODOPL. 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 VSN).</td> </tr> </tbody> </table>	Value	Description	NO	Output program libraries are not generated or written.	YES	The installation procedures write output program libraries. This does not apply to MODOPL. 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 VSN).
Value	Description						
NO	Output program libraries are not generated or written.						
YES	The installation procedures write output program libraries. This does not apply to MODOPL. 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 VSN).						
NOTE							
<p>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:</p> <pre> OUTPRD=NO PSRIN=nnn PSROUT=nnn </pre> <p>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.</p>							
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 an asterisk (*). This parameter is significant only if IUCHG=YES.						
PFGPN	Called from common deck COMPFG. Specifies the auxiliary pack name where PFGxxxx files are written. The default option is to write these files to the catalog under which the installation procedure is executing.						
PFGPR	Called from common deck COMPFG. Specifies the auxiliary pack type where PFGxxxx files are written when the files are to reside on an auxiliary pack.						
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.						

Parameter	Description						
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 overwrite the input PLs.						
RESOUR	Called from common deck COMIUN. Specifies the format of the RESOURC command to use; for example: *N=\$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=PRINT.						
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>PRINT</td> <td>Job output is printed at the central site.</td> </tr> <tr> <td>WAIT</td> <td>Job output (everything on file OUTPUT) is placed in the wait queue with a user job name of the same name as the installation procedure name followed by a B or an F. (If the procedure name is seven characters, the last letter of the name is replaced by the B or F.) If the job passed, the last letter of the output file name is B; if the job failed, the last letter of the output file name is F.</td> </tr> </tbody> </table>	Value	Description	PRINT	Job output is printed at the central site.	WAIT	Job output (everything on file OUTPUT) is placed in the wait queue with a user job name of the same name as the installation procedure name followed by a B or an F. (If the procedure name is seven characters, the last letter of the name is replaced by the B or F.) If the job passed, the last letter of the output file name is B; if the job failed, the last letter of the output file name is F.
Value	Description						
PRINT	Job output is printed at the central site.						
WAIT	Job output (everything on file OUTPUT) is placed in the wait queue with a user job name of the same name as the installation procedure name followed by a B or an F. (If the procedure name is seven characters, the last letter of the name is replaced by the B or F.) If the job passed, the last letter of the output file name is B; if the job failed, the last letter of the output file name is F.						
TOLIST=option	Called from common deck COMTOL. Specifies the assembly list file routing and whether the file specified by the LIST parameter goes to the wait queue. The default is TOLIST=NONE; if the job fails, the list file is discarded.						
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NONE</td> <td>The list file, if defined, is local to the job and is discarded when the job terminates.</td> </tr> <tr> <td>WAIT</td> <td>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.</td> </tr> </tbody> </table>	Value	Description	NONE	The list file, if defined, is local to the job and is discarded when the job terminates.	WAIT	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.
Value	Description						
NONE	The list file, if defined, is local to the job and is discarded when the job terminates.						
WAIT	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.						
	<p>NOTE</p> <hr/> <p>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.</p> <hr/>						

Parameter	Description						
UN1	<p>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.</p> <p>NOTE</p> <p>If you specify a user name for UN1, DISKINS must be set to NO if the source program libraries have not been preloaded.</p>						
UN2	<p>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.</p> <p>NOTE</p> <p>If you specify a user name with the UN2 parameter, OUTPRD must be set to NO.</p>						
USEPFG	<p>Called from common deck COMPFG. Specifies whether you want to save the PFGxxxx files created for use with a SYSGEN installation procedure. The default is USEPFG=NO. Residence of these files is controlled by the COMMOD parameters PFGPN and PFGPR.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO</td> <td>Only PFGxxxx files that previously existed are replaced with current PFGxxxx files. If a file does not exist, no new permanent file is created.</td> </tr> <tr> <td>YES</td> <td>All PFGxxxx files are saved regardless of whether they existed previously.</td> </tr> </tbody> </table>	Value	Description	NO	Only PFGxxxx files that previously existed are replaced with current PFGxxxx files. If a file does not exist, no new permanent file is created.	YES	All PFGxxxx files are saved regardless of whether they existed previously.
Value	Description						
NO	Only PFGxxxx files that previously existed are replaced with current PFGxxxx files. If a file does not exist, no new permanent file is created.						
YES	All PFGxxxx files are saved regardless of whether they existed previously.						

Parameter	Description						
USERF	Called from common deck COMUSER. Specifies that there is a USER file for this installation procedure. The default value is null. This common deck is not used by the CCP/CROSS installation procedures. Refer to chapter 7 for information on USER files for CCP/CROSS.						
	<table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>null</td> <td> <p>A USER file is searched for only if a site includes the USERF parameter on the installation procedure call. For example:</p> <pre>BEGIN, AAM2, INSTALL, USERF=AAM2MOD.</pre> <p>If the USER file (in this case AAM2MOD) is not found, the installation procedure aborts. The site may choose any file name to be the USER file. If you specify USERF=UJOBNAM on the installation procedure call, it overrides the null value.</p> </td> </tr> <tr> <td>UJOBNAM</td> <td> <p>A USER file is searched for automatically. The file name that is looked for is a U concatenated with the first six characters of the installation procedure name (for example, UFTN5 or UTERMLI). If no file with that name is found, the installation procedure executes as usual and no user code is applied. If you specify USERF=filename on the installation procedure call, it overrides the UJOBNAM value.</p> </td> </tr> </tbody> </table>	Value	Description	null	<p>A USER file is searched for only if a site includes the USERF parameter on the installation procedure call. For example:</p> <pre>BEGIN, AAM2, INSTALL, USERF=AAM2MOD.</pre> <p>If the USER file (in this case AAM2MOD) is not found, the installation procedure aborts. The site may choose any file name to be the USER file. If you specify USERF=UJOBNAM on the installation procedure call, it overrides the null value.</p>	UJOBNAM	<p>A USER file is searched for automatically. The file name that is looked for is a U concatenated with the first six characters of the installation procedure name (for example, UFTN5 or UTERMLI). If no file with that name is found, the installation procedure executes as usual and no user code is applied. If you specify USERF=filename on the installation procedure call, it overrides the UJOBNAM value.</p>
Value	Description						
null	<p>A USER file is searched for only if a site includes the USERF parameter on the installation procedure call. For example:</p> <pre>BEGIN, AAM2, INSTALL, USERF=AAM2MOD.</pre> <p>If the USER file (in this case AAM2MOD) is not found, the installation procedure aborts. The site may choose any file name to be the USER file. If you specify USERF=UJOBNAM on the installation procedure call, it overrides the null value.</p>						
UJOBNAM	<p>A USER file is searched for automatically. The file name that is looked for is a U concatenated with the first six characters of the installation procedure name (for example, UFTN5 or UTERMLI). If no file with that name is found, the installation procedure executes as usual and no user code is applied. If you specify USERF=filename on the installation procedure call, it overrides the UJOBNAM value.</p>						
VARLIST	Called from common deck COMCCPV. Used by CCP/CROSS installation procedures only (refer to chapter 7).						
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, REP=n, MOD=fn, EXPAND=nn, MODIF, UMODE.
```

NOTE

Include all the keyword=value parameters before using the keyword parameters.

Parameter	Description
EXPAND=nn	Specifies whether common deck calls are expanded. You can specify YES or NO. The default is YES.
MOD=fn	Specifies the name of a modification set file to add to the PL for listing purposes. If the file is local, a permanent file of the same name is ignored. The PL is not permanently updated.
MODIF	Includes the default Modify list options on the listing.
REP=n	Specifies the number of additional copies to be printed. The default is 0.
UMODE	Lists only modified decks; used in conjunction with the MOD=fn parameter.

Here are some examples of calls to the DECKLIS procedure:

```
BEGIN, DECKLIS, INSTALL, MOD=COMMOD, UMODE.
```

```
BEGIN, DECKLIS, INSTALL, EXPAND=NO.
```

Disk Installations

If you want to perform your installation with the source program libraries on disk, which allows 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 load them one at a time as they are needed. If you want to preload your files, refer to Using PFLOAD, later in this section.

Each installation procedure 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. If the file is not on disk, PRDIN issues a RECLAIM command to load the file from the permanent file tapes and then stores 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 at PSR level 999 would be NAM5999 (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 (OUTPRD=NO) 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 (OUTPRD=YES), you must change the value of PSROUT so that it is different from the value of PSRIN.

The parameters OUTPRD, PSRIN, and PSROUT are common parameters. Refer to COMMOD File Parameters, earlier in this chapter.

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 should specify the OP=Z and UD parameters since Control Data maintains these files using Mass Storage Extended (MSE).
- To force PFLOAD to load files to the proper user index, you must specify the DI parameter.
- If you are loading a family device, you must specify the FM and DD parameters.
- If you are loading an auxiliary device, you must specify the PN parameter.
- If you are loading multiple disk packs, you might want to use the OP=L option to cause PFLOAD to perform load leveling.
- For more information about PFLOAD, refer to the NOS Version 2 Analysis Handbook.

All source program libraries are loaded with names of the format xxxpsrin where xxxx is the 4-character product name (such as TEXT or NAM5) and psrin is the value of the PSRIN parameter which, by default, is the current release level number.

Thus, the source program library for NAM5 at PSR level 999 would be loaded with the name NAM5999. The name for the composite OPL is OPLpsrin; thus, for the release at PSR level 999, the name would be OPL999.

All other files are given names using the format PFGxxxx, where xxxx is the 4-character product name.

The PFGxxxx files are used by SYSGEN. Refer to the SYSGEN command in chapter 8 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 null. Each product source file is assigned to one of the common decks named COMD1, COMD2, COMD3, COMD4, or COMXOPL. 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 9-1.

Each of the common decks contains 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 installation procedures that use the auxiliary program library. This second deck describes the auxiliary program library disk location. It is named using the following format:

COMXa

Parameter	Description
COMX	Identifies an auxiliary common deck in DECKOPL.
a	One-character, identical to the last character of the common deck used by PRDIN to locate the input product file.

Example:

The RHC product file has its disk assignment in COMD3. Product RHF uses RHC as an auxiliary program library and, thus, includes the deck COMX3. The pack assignment could be as follows:

```

COMD3          COMX3

PN=PACKY
PR=DJ
PN0=PACKX     PN3=PACKX
PRO=DJ        PR3=DJ

```

Table 9-1. Source File and Auxiliary PL Assignments

Common Deck	Installation Procedure	Auxiliary Common Deck¹	Source File	Required Aux PLs²
COMD1	AAM2	COMX1	AAM2	
COMD1	APL2		APL2	X
COMD1	BAM		BAM1	1
COMD1	BASIC3		BAS3	X,1
COMD1	BIT8		BIT8	
COMD1	CCG	COMX1	CCG1	
COMD1	CDCS2		CDCS	X,1
COMD1	CID		CID1	X
COMD1	COBOL5		COB5	
COMD1	COMPASS	COMX1	CPS1	
COMD1	DCL		DCL2	
COMD1	DDL3	COMX1	DDL3	1
COMD1	FCL1		FCL4	1
COMD1	FCL2		FCL4	1
COMD1	FCL5		FCL5	1
COMD1	FDBF		FDBF	1
COMD1	FORM		FORM	
COMD1	FTN4		FTN4	1
COMD1	FTNTS		FTI4	1
COMD1	FTN5		FTN5	1
COMD1	F45		F451	1
COMD1	LOADER		LDR1	
COMD1	PASCAL		PASC	X
COMD1	PMD		PMD5	1
COMD1	QU3		QU31	X,1
COMD1	SORT5		SRT5	
COMD1	SYMPL		SYMP	
COMD1	TEXT	COMX1	TEXT	
COMD1	TEXTIO		TXIO	
COMD1	UPDATE		UPD1	1
COMD2	TCPH		TCPH	X,1,2
COMD2	CHA1		CHA1	X,2
COMD2	MCS		MCS1	X,2
COMD2	NAM5	COMX2	NAM5	X
COMD2	PSU		PSU1	X
COMD2	RBF5		RBF5	X,2

1. A common deck name in this column identifies the source file as one also used as an auxiliary PL.

2. Each letter or digit in this column refers to common decks for auxiliary PLs needed by this installation procedure. For example, X means COMXOPL, 1 means COMX1, 2 means COMX2, and so forth.

(Continued on next page)

Table 9-1. Source File and Auxiliary PL Assignments

Common Deck	Installation Procedure	Auxiliary Common Deck¹	Source File	Required Aux PLs²
<i>(Continued from previous page)</i>				
COMD3	AP1I		AP1I	
COMD3	AP1L		MMCL	
COMD3	BINEDIT		BINE	1
COMD3	CCL		CCL1	X,1
COMD3	CEDIAG		CEDG	X
COMD3	DUAL		DUAL	X,1
COMD3	FCS3		FCS3	
COMD3	FORMAT		FMAT	X
COMD3	HCD		ZHCD	
COMD3	ITF		ITF1	X,3
COMD3	LCS3		LCS3	
COMD3	MMCL		MMCL	
COMD3	MODOPL	COMXOPL	OPL	
COMD3	MSSI		MSSI	X
COMD3	NSS		NSS1	X
COMD3	PFTF		PFTF	X
COMD3	RHC	COMX3	RHC1	
COMD3	RHF		RHF1	X,3
COMD3	RHP		RHP1	X,3
COMD3	TDU		TDU1	X
	FSE			X
	HSIO			X
	IAF			X
	MMF			X
	MSE			X
	NIP5870			X
	NOS			X
	NOS2B			X
	OSLIB			X
	OSTEXT			X
	RDFEX			X
	TAF			X
	TOOLS			X
	TRACER			X
	XEDIT			X

1. A common deck name in this column identifies the source file as one also used as an auxiliary PL.

2. Each letter or digit in this column refers to common decks for auxiliary PLs needed by this installation procedure. For example, X means COMXOPL, 1 means COMX1, 2 means COMX2, and so forth.

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										
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: <table border="1" data-bbox="500 653 1365 821"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>HY</td> <td>800 cpi (7-track).</td> </tr> <tr> <td>HD</td> <td>800 cpi (9-track).</td> </tr> <tr> <td>PE</td> <td>1600 cpi (9-track).</td> </tr> <tr> <td>GE</td> <td>6250 cpi (9-track).</td> </tr> </tbody> </table>	Value	Description	HY	800 cpi (7-track).	HD	800 cpi (9-track).	PE	1600 cpi (9-track).	GE	6250 cpi (9-track).
Value	Description										
HY	800 cpi (7-track).										
HD	800 cpi (9-track).										
PE	1600 cpi (9-track).										
GE	6250 cpi (9-track).										
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.										
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.										
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.										

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 to make these additions. (For information on LIBEDIT directives, refer to NOS Version 2 Reference Set, Volume 3.)

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

Code that may correct a specific user site problem but that has not been fully tested is released on the file MISCPL. This file, if it exists, 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 the 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 of the deck 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. + .modname1 + . . . + .modnamen
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 installation procedure.

REPORT Procedure

To obtain statistics on all completed installation procedures, run a job that contains this command. The job output indicates the resources used for each installation procedure and whether the procedure passed or failed.

```
BEGIN, REPORT, INSTALL, XC .
```

NOTE

The REPORT procedure uses the direct access file REP which is the FTN5 binary that actually performs the resource calculation. If the binary file cannot be found or if the XC keyword is present, REP is recompiled prior to generating the report.

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 installation procedures, and creates DIRFILE with entries for those ULIBs. The installation procedures subsequently add binaries to the file PRODUCT and directives to file DIRFILE. Only those user libraries used by subsequent installation procedures 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(SOURCE) function initially loads the permanent files needed for customizing an installation. Refer to Loading Permanent Files in chapter 8 for a complete list. 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, NEWPL, MOD=filename, DF63, CV63, INSTALL=filename.
```

Parameter	Description
CV63	Converts DECKOPL to 63-character set format.
DF63	Selects 63-character set version of installation procedures for inclusion on file INSTALL.
INSTALL=filename	Creates or replaces the procedure file specified. The default is INSTALL. If you omit the INSTALL keyword, procedure file INSTALL is not created or replaced.
MOD=filename	Applies modsets from the specified file to DECKOPL; the file can be a local file or a 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. If MOD is omitted, no modifications are applied.
NEWPL	Replaces DECKOPL with a modified DECKOPL. If NEWPL is omitted, DECKOPL is not 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.
```


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.

Batch Critical Update (BCU)

A binary correction to CDCNET or Dual State.

Batch Origin Job

A job submitted from a batch terminal.

BCU

Refer to Batch Critical Update.

CCP

Refer to Communications Control Program.

CDCNET

Refer to Control Data Distributed Communications Network.

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

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.

CM

Refer to Central Memory.

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.

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 Control Program (CCP)

Software product used to control 255x Network Processing Units.

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 in a NAM/CCP network.

Control Data Distributed Communications Network (CDCNET)

The collection of compatible hardware and software products offered by Control Data Corporation to interconnect computer resources into distributed communications networks and that is compatible with Control Data Network Architecture (CDNA).

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 database.

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.

DAS

Refer to Disk Array Subsystem.

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 some 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.

DC

Refer to Disposition Code.

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

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.

Device Interface (DI)

CDCNET hardware for open system interconnections. The device interface houses processor boards in configurations that permit a network of various other data processing equipment.

DI

Refer to Device Interface.

Direct Access File

A NOS permanent mass storage file accessed by attaching the original file 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

Refer to Job Display.

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 Array Subsystem

A mass storage subsystem.

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. The most frequently used are the job status (B), system and local 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

Refer to Dynamic System Display.

Dual State

The state in which two operating systems execute simultaneously on the same mainframe. NOS/VE and either NOS Version 2 or NOS/BE Version 1.5 are such partners.

Dynamic System Display (DSD)

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

Refer to Extended Core Storage and Extended Memory.

EJT

Refer to Executing Job Table.

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

The number from 0 to 7 that 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.

ESM

Refer to Extended Semiconductor Memory.

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.

Executing Job Table (EJT)

A central memory resident table that contains a 4-word entry for all executing jobs, including interactive service class jobs. It is used to control jobs that are executing at a

control point and jobs that are rolled out. Every executing job in the system has an EJT entry.

Extended Core Storage (ECS)

A type of extended memory that is an option available for 6000 Computer Systems, CYBER 70 Computer Systems, CYBER 170 Computer Systems (except model 176), and CYBER 180 Computer Systems. The maximum size of ECS is two million words. ECS is equipped with a CPU port and optionally a distributed data path (DDP). (The CPU port cannot be connected to a CYBER 180-class mainframe.) See Extended Memory.

Extended Memory (EM)

An additional portion of memory available as an option. This memory can be used for program and data storage, but not for program execution. Special hardware instructions exist for transferring data between central memory and extended memory. Extended memory consists of either extended core storage (ECS), extended semiconductor memory (ESM), large central memory extended (LCME), unified extended memory (UEM), or STORNET.

Extended Semiconductor Memory (ESM)

A type of extended memory that is an option available for 6000 Computer Systems, CYBER 70 Computer Systems, CYBER 170 Computer Systems (except model 176), and CYBER 180 Computer Systems. The maximum size of ESM is 16 million words. ESM is equipped with a CPU port and one or more low-speed ports (LSP). (The CPU port cannot be connected to a CYBER 180-class mainframe.) See Extended Memory.

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.

File Transfer Protocol (FTP)

The Control Data application-to-application protocol that enables applications programs executing on one host system to exchange information with applications programs that execute on other host systems.

FL

Refer to Field Length.

FOT

Refer to Family Ordinal Table.

FTP

See File Transfer Protocol.

IAF

Refer to Interactive Facility.

Indirect Access File

A NOS permanent file accessed 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.

Intelligent Peripheral Interface

An industry supported computer interface for high performance peripherals developed by the American National Standards Institute.

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.

IPI

Refer to Intelligent Peripheral Interface.

Job Display (DIS)

A system peripheral processor program similar to the 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.

Job Sequence Name (JSN)

The unique system-defined name assigned to every executing job or queued file. The JSN is a string of 4 alphabetic 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, the job status indicates why it was rolled out.

JSN

Refer to Job Sequence Name.

Large Central Memory Extended (LCME)

A type of extended memory that is an option available for CYBER 170 model 176. See Extended Memory.

LCF

Refer to Local Configuration File.

LCME

Refer to Large Central Memory Extended.

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 lists the application programs available for execution in the host computer and the users who 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.

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.

Mainframe Device Interface (MDI)

The standard CDCNET DI variant that interconnects a CYBER 170/180 mainframe with an Ethernet local area network.

Mainframe Terminal Interface (MTI)

The standard CDCNET DI variant that interconnects CYBER 170/180 host computers with terminals, workstations, and unit record equipment without requiring a local area network.

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.

MDI

Refer to Mainframe Device Interface.

MID

Refer to Machine Identifier.

MST

Refer to Mass Storage Table.

MTI

Refer to Mainframe Terminal Interface.

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, DIs, 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 a NAM/CCP 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 Load File (NLF)

A file containing CCP software to load into a 255x Network Processing Unit (NPU).

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 NAM application program, that dumps and loads network processing units (NPUs) upon request in a NAM/CCP network.

Network Validation Facility (NVF)

A portion of the network software, written as a NAM application program, that performs application validation and all connection validation processing and supports login dialogue with the terminal user.

NLF

Refer to Network Load File.

NPU

Refer to Network Processing Unit.

NS

Refer to Network Supervisor.

NVF

Refer to Network Validation Facility.

Order-Dependent

Used to describe items that must appear in a specific order, such as parameters.

Order-Independent

Used to describe items that 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 output 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 microcode 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 multmainframe 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. The system protects the file from unauthorized access according to privacy controls specified when the file was 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.

Printer Support Utility (PSU)

Operating system software used to control the 533/536/585 printers.

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.

PSU

Refer to Printer Support Utility.

QFT

Refer to Queued File Table.

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.

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 Origin 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.

SC

Refer to Service Class.

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.

Service Class (SC)

An attribute associated with a queued file or executing job. Service class determines how the system services the job.

SSD

Solid State Disk.

STORNET

STORNET

A type of extended memory that is an option available for CYBER 170 Computer Systems (except model 176) and CYBER 180 Computer Systems. The maximum size of STORNET is 4 million words. STORNET is equipped with one or more low-speed ports (LSP), but is not equipped with a CPU port. See Extended Memory.

Suspended Job

An interactive job placed in an inactive state. Processing stops immediately and recovery information is copied to the rollout file. If the job's EJT entry is recovered, processing resumes as if no interruption took place.

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 that are needed to run the operating system and its product set.

System Origin Job

A job entered at the system console.

TCP/IP

Refer to Transmission Control Protocol/Internet Protocol.

TCU

Refer to Trunk Control Unit.

TDI

Refer to Terminal Device Interface.

Terminal Device Interface (TDI)

The standard CDCNET DI variant that connects terminals, workstations, and unit record equipment to a local area network.

Transmission Control Protocol/Internet Protocol (TCP/IP)

A suite of protocols that supports the Defense Advanced Research Projects Agency Network (ARPANET) activities. TCP/IP must be implemented within CDCNET for connectability to Defense Data Networks (MILNET or ARPANET) and to workstations that use TCP/IP.

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.

UEM

Refer to Unified Extended Memory.

UJN

Refer to User Job Name.

Unified Extended Memory (UEM)

A type of extended memory that is available as an option for CYBER 180-class machines and models 865 and 875. UEM differs from other types of extended memory in that it is a portion of central memory and not a separate memory unit. See Extended Memory.

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.

File Formats

B

Order of Products B-1

Permanent File Tapes B-3

Order of Products

The order in which products appear on the deadstart tape is now controlled by DECKOPL in each installation procedure. The basic order is alphabetical.

NOTE

All ULIBs are in library 4.

Arrangement of deadstart tape libraries:

Library	Description	Contributing Installation Procedures
1	CTI (Fixed order)	
2	(Fixed order)	NOS, HSIO, MMF, Disk Microcode
3	NOS	NOS, NOS2B
4	All ULIBs	Many
5	Miscellaneous items	TDU, PFTF, NIP5870, Misc. Microcode
6	AAM2	AAM2
7	APL2	APL2
8	ATF	ATF
9	BAM1	BAM
10	BAS3	BASIC3
11	BINE	BINEDIT
12	BIT8	BIT8
13	CCL1	CCL
14	CDCS	CDCS2
15	CEDG	CEDIAG
16	CHA1	CHA1
17	CID1	CID
18	CML1	
19	COB5	COBOL5/COBOL5Q
20	CPS1	COMPASS
21	DDL3	DDL3
22	DUAL	DUAL
23	FDBF	FDBF
24	FORM	FORM
25	FMAT	FORMAT
26	FSE1	FSE
27	FTN4	FTN4/FTNTS, FCL1, FCL2
28	FTN5	FTN5, FCL5
29	F451	F45
30	IAF1/RDF1	IAF, RDFEX
31	ITF1	ITF
32	LDR1	LOADER
33	MAP	MMCL, MSSI, APII
34	MCS1	MCS
35	(empty)	

Library	Description	Contributing Installation Procedures
36	MMF1	MMF
37	MSE1	MSE
38	(empty)	
39	NAM5	NAM5/NAM5D
40	(empty)	
41	NSS1	NSS
42	OSTX	OSTEXT
43	PASC	PASCAL
44	(empty)	
45	PMD5	PMD
46	PSU1	PSU
47	QU31	QU3
48	RBF5	RBF5/RBF5D
49	RHF1	RHF
50	(empty)	
51	RHP1	RHP
52	SRT5	SORT5
53	SYMP	SYMPL
54	TAF1	TAF
55	TCP/IP/FTP/TELNET	TCPH
56	TEXT	TEXT
57	TXIO	TEXTIO
58	TMS	TMS
59	TOOL	TOOLS
60	TRCE	TRACER
61	UPD1	UPDATE
62	XEDT	XEDIT
63	(empty)	

Permanent File Tapes

This section lists all files on the permanent file tapes. All source files are listed first, followed by all PFGxxxx files (files processed by SYSGEN). At the end of the list are files received only on a component order.

NOTE

Your tape contains only the files for the products you have ordered.

To see a list of all the files on your permanent file tapes, follow these steps:

1. Load the RECLAIM database as described under Loading Permanent Files in chapter 8.
2. Enter this command from your installation user name:

```
RECLAIM,Z./LIST,UN=NS2psrin
```

Replace psrin with the release level.

This generates a report that lists the names of all the files on the tapes in alphabetical order. Because the report is generated from information in the RECLAIM database, the permanent file tapes are not requested.

All source files end with the value psrin, the current NOS release level. For example, the AAM2 source file would appear as AAM2999 at NOS level 999. The table does not list the psrin value.

Unless otherwise noted, all source files contain one random formatted Update program library.

File Name	Associated DECKOPL Procedure Name	Format Notes
AAM2psrin	AAM2	
APL2psrin	APL2	File 1. Program library in Modify format. 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.

File Name	Associated DECKOPL Procedure Name	Format Notes
AP1Ipsrin	AP1I	AP1 online diagnostics for MAP III and MAP IV. File 1. Sequential PL in Update format. File 2. CSTFU. File 3. CSCMD. File 4. CSQMM. File 5. CSMAINT. File 6. CSVDMT. File 7. CS24KM. File 8. CSDSOP4. File 9. CSECS. File 10. CSECSTA. File 11. CSHSMT for 24K memory size. File 12. CSHSMT for 48K memory size. File 13. CSHSMT for 64K memory size. File 14. CSCMI.
BAM1psrin	BAM	
BAS3psrin	BASIC3	
BINEpsrin	BINEDIT	
BIT8psrin	BIT8	
CCG1psrin	CCG	
CCL1psrin	CCL	
CCPBpsrin	CCPPH1, CCPBLB	File 1. CCP base – sequential PL in Update format. 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.
CCPDpsrin	CCPPH1	CCP Online Diagnostics – sequential PL in Update format.
CCPRpsrin	CCPPH1	CCP Remote Concentrator Products (LIP) – sequential PL in Update format.
CCPTpsrin	CCPPH1	CCP 3270 Bisync TIP – sequential PL in Update format.
CDCSpsrin	CDCS2	
CEDGpsrin	CEDIAG	
CHA1psrin	CHA1	

File Name	Associated DECKOPL Procedure Name	Format Notes
CID1psrin	CID	
COB5psrin	COBOL5/COBOL5Q	File 1. Sequential PL in Update format. Files 2,3,4. Used by the COBOL5Q installation procedure.
CPS1psrin	COMPASS	
CRSSpsrin	CROSS	File 1. Sequential PL in Update format. 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.
DCL2psrin	DCL	File 1. Sequential PL in Update format. 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.
DDL3psrin	DDL3	
FCL4psrin	FCL1, FCL2	
FCL5psrin	FCL5	
FCS3psrin	FCS3	File 1. Sequential PL in Update format. 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.

File Name	Associated DECKOPL Procedure Name	Format Notes
FDBFpsrin	FDBF	
FMATpsrin	FORMAT	
FORMpsrin	FORM	
FTI4psrin	FTNTS	
FTN4psrin	FTN4	
FTN5psrin	FTN5	
F451psrin	F45	
ITF1psrin	ITF	
LCS3psrin	LCS3	File 1. Sequential PL in Update format. 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).
LDR1psrin	LOADER	
MCS1psrin	MCS	
MMCLpsrin	MMCL, AP1L	Sequential PL in Update format.
MSSIpsrin	MSSI	Sequential PL in Update format.
NAM5psrin	NAM5/NAM5D	
NSS1psrin	NSS	
OPLpsrin		This is a composite OPL of all the NOS Modify products you ordered. It may contain the PLs for NOS, FSE, HSIO, IAF, MMF, MSE, TAF, TOOLS, TRACER, and XEDIT. OPLpsrin also contains the input PLs for the installation procedures NIP5870, NOS2B, OSLIB, OSTEXT, RDFEX, TDU, and TERMLIB.
PASCpsrin	PASCAL	
PFTFpsrin	PFTF	Program library in Modify format.
PMD5psrin	PMD	
PSU1psrin	PSU	
QU31psrin	QU3	
RBF5psrin	RBF5/RBF5D	
RHC1psrin	RHC	
RHF1psrin	RHF	
RHP1psrin	RHP	
SRT5psrin	SORT5	
SYMPpsrin	SYMPL	
TCPHpsrin	TCPH	
TDU1psrin	TDU	File 1. TDUTOOL 1. File 2. TDUTOOL 2. File 3. TDUTOOL 3.

File Name	Associated DECKOPL Procedure Name	Format Notes
TEXTpsrin	TEXT	
TXIOpsrin	TEXTIO	
UPD1psrin	UPDATE	
VSM1psrin	CCPVAR	CCP variant SM1 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).
VSM2psrin	CCPVAR	CCP variant SM2; same format as file VSM1.
VSM3psrin	CCPVAR	CCP variant SM3; same format as file VSM1.
VSM4psrin	CCPVAR	CCP variant SM4; same format as file VSM1.
VV1Fpsrin	CCPVAR	CCP variant V1F; same format as file VSM1.
VV1Lpsrin	CCPVAR	CCP variant V1L; same format as file VSM1.
VV2Fpsrin	CCPVAR	CCP variant V2F; same format as file VSM1.
VV2Lpsrin	CCPVAR	CCP variant V2L; same format as file VSM1.
VV3Fpsrin	CCPVAR	CCP variant V3F; same format as file VSM1.
VV3Lpsrin	CCPVAR	CCP variant V3L; same format as file VSM1.
VV4Fpsrin	CCPVAR	CCP variant V4F; same format as file VSM1.
VV4Lpsrin	CCPVAR	CCP variant V4L; same format as file VSM1.
ZHCDpsrin	HCD	819 PP Driver.

All of the following files are processed by SYSGEN. They are stored on the permanent file tapes 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	File Format/Notes
PFGAAAA		LOADUSE	SYSGEN data file.
PFGAPL2	APL2	APL2	File 1. Empty file. File 2. APPLIB. Relocatable of APL. File 3. APLPROD. Absolute of APL. File 4. TAPLTST. APL verification 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 verification. File 13. Reserved.
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.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	File Format/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.
PFGBCU1			File 1. Installation procedures. File 2. RECLAIM database. File 3. RECLAIM dump of CDCNET permanent files.
PFGBCU2			File 1. Dual State Source Library in NOS/VE format. File 2. VEMEM procedures. File 3. RECLAIM dump of dual state binaries for earlier NOS releases.
*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	CCP NLFFILE.
*PFGCCPU		CCPU	CCP USERBPS file.
PFGDCS	CDCS2	CDCS	CDCS startup procedure.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	File Format/Notes
PFGCHA1	CHA1	CHA1	File 1. Installation procedure. File 2. CHA1 message template file, HTF_id_psroun. File 3. CHA1 response file, HRF_id_psroun. File 4. CHA1 log file, HLF_id_psroun. File 5. CHA1 NAMSTRT records.
PFGCHA2		CHA2	File 1. Installation procedure. File 2. NETOU template file, TF_id_vvvv.
PFGCML1		CML1	Refer to the CML Reference Manual for specific file format information.
*PFGCODE		CODE	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 6. MPLINK. File 7. MPEDIT. File 8. CCP Pascal compiler. File 9. PASCAL compiler bootstrap.
PFGCSTD		CSTD	File 1. COMPASS coding standards. File 2. SYMPL coding standards.
*PFGDBU1		DBU1	DBUBIN. (For use by QU3.)
PFGDCL2	DCL	DCL2	Permanent files for Data Catalogue 2 Version 2.0. 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.
PFGDCNS		DCNS	File 1. Installation procedures. File 2. RECLAIM database. File 3. RECLAIM dump of CDCNET permanent files.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	File Format/Notes
*PFGDECK		DECK	DECKOPL permanent files. File 1. DECKOPL. File 2. INSTALL procedure file. File 3. COMMOD. File 4. GDIR program binaries.
PFGDUAL	DUAL	DUAL	File 1. Dual State Source Library in NOS/VE format. File 2. VEMEM procedures. File 3. RECLAIM dump of dual state binaries for earlier NOS releases.
PFGFSE1	FSE	FSE1/FSEH/ FSES	File 1, record 1. Contains SMF startup procedure. File 1, record 2. FSEHELP file. File 1, record 3. FSTEACH file. File 1, record 4. FSEPROC file. File 1, record 5. SMFSTAT file.
PFGIAF1	IAF	IAF1	Interactive Facility Startup procedures. File 1, record 1. IAF startup procedure. File 1, record 2. IAFTM startup procedure. File 1, record 3. IAFTR startup procedure.
PFGITF1	ITF	ITF1	ITF NAMSTRT startup record.
PFGMAN1		MAN1	Optional set of online manuals: procedures, source and bound copies of the manuals. File 1. Contains the installation procedure and documentation for all other files within PFGMAN1.
PFGMCS1	MCS	MCS1	MCS startup procedure.
*PFGMISC		MISC	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.
PFGMSE1	MSE	MSE1	File 1, record 1. MSE startup procedure. File 1, record 2. MSE slave startup procedure. File 1, record 3. MSE sample configuration file.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	File Format/Notes
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. Sequential PL in Update format.
PFGNAMD	NAM5D	SWAP	NAM5 trace/debug binaries. SYSGEN procedure SWAP is used to load these binaries.
PFGNAM5	NAM5	NAM5	NAM startup files. File 1. NAMSTRT file. File 2. NAM startup procedure. File 3. NAMNOGO startup procedure.
PFGNDL1		NDL1	Network Definition File. File 1. NDL source file NDLDATA. File 2. Compiled NCFFILE of NDLDATA. File 3. Compiled LCFFILE of NDLDATA.
PFGNIPB		SWAP	63-character set NIP5870 binaries. SYSGEN procedure SWAP is used to load these binaries.
PFGNOSB	NOS2B	NOSB/HELP	HELPLIB file.
PFGNOS2	NOS	NOS2	STAGE job startup procedure.
PFGNSS1	NSS	NSS1	NOS Scope Station startup procedure.
PFGONLM		ONLM	Default set of online manuals: procedures, source and bound copies of the manuals. File 1. Contains the installation procedure and documentation for all other files within PFGONLM.
PFGPFTF	PFTF	PFTF	Protocol File Transfer Facility. File 1. Procedure RMUGET (64-character set version). File 2. Procedure RMUGET (63-character set version). File 3. User library PFTF (debug version).

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	File Format/Notes
*PFGPRPT		PRPT	History idents of NOS PSR code in current release.
PFGPSU1	PSU	PSU1	File 1. PSU NAMSTRT startup record. File 2. PSU default settings file, EVFULFN.
PFGRBFD	RBF5D	SWAP	RBF5 trace/debug binaries. SYSGEN procedure SWAP is used to load these binaries.
PFGRBF5	RBF5	RBF5	RBF NAMSTRT startup record.
PFGRDF1	RDFEX	RDF1	RDF startup procedure.
PFGRHP1	RHP	RHP1	RHP NAMSTRT startup record.
PFGTAF1	TAF	TAF1	Transaction Facility permanent files. File 1. TAF startup procedure. File 2. TASKLIB.
PFGTCPH	TCPH	TCPH	TCP/IP/FTP/TELNET NAMSTRT startup records.
PFGTOOL	TOOLS	TOOL	Maintenance tools files. File 1. SECART – Security audit reduction tool. File 2. MSGID – Security audit message file.
PFGTLIB	TERMLIB	TLIB	Terminal Definition Utility files. File 1. User library TERMLIB. File 2. TDUFILE.
PFGXEDT	XEDIT	XEDT	XEDIT help file, XEDITH.

On component orders, you also get the following files:

File Name	Notes
BLDLIB	Contains a procedure to properly update all ULIBs from binaries in file PRODUCT.
DSTDIR	Contains LIBEDIT directives for the products in file PRODUCT.
PRODUCT	Contains all binaries for the deadstart tape.

63-Character Set Installation

C

Installation Procedure	C-1
File Conversions	C-2
5870 Installation	C-2
Protocol File Transfer Facility (PFTF)	C-2

This appendix describes how to install a system with a 63-character set format.

Installation Procedure

To install a 63-character set system, follow these steps:

1. Load the files necessary for installation (refer to chapter 6).
2. Execute the following command to convert the Modify-formatted DECKOPL procedures to 63-character set format:

```
BEGIN, SETUP, INSTALL, CV63, NEWPL.
```

3. Execute the following command to create an INSTALL procedure file in 63-character set format which includes code for a 63-character set installation:

```
BEGIN, SETUP, INSTALL, DF63, INSTALL.
```

NOTE

Any future calls to SETUP should include the DF63 parameter.

4. Run the UCOMMODO procedure to recreate file COMMODO. Modify COMMODO parameters, if necessary.

```
BEGIN, UCOMMODO, INSTALL.
```

Use the new COMMODO file to set up DECKOPL installation defaults.

```
BEGIN, SETUP, INSTALL, DF63, INSTALL, MOD=COMMODO.
```

5. Run the SEED procedure. Refer to chapter 6 for more information.
6. Create a USER file for the MODOPL procedure, if needed.
7. Execute the MODOPL procedure. The composite OPL is created in 63-character set format. Refer to chapter 6 for more information.
8. If you are installing on a dedicated system, deadstart the system again using this IPRDECK entry:

```
CSM=63.
```

If you are installing on a production system, skip this step.

9. Include the parameter CSET=C63 or CSET=A63 on the call to build TEXT.
10. Continue with the installation.

File Conversions

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, to convert from display code in 64-character set to ASCII code in 63-character set, use this command:

```
FCOPY(P=oldfile,PC=DIS64,N=newfile,NC=ASCII63,R)
```

To convert from a 64-character set ASCII code to a 63-character set ASCII code, use this command:

```
FCOPY(P=oldfile,PC=ASCII64,N=newfile,NC=ASCII63,R)
```

The following files (on the installation user name) are installed by SYSGEN and must be converted to 63-character set format because they are not recreated from the installation procedures:

- USERBPS
- NDLDATA
- PSRRPT
- SRB (ASCII 6/12)

5870 Installation

On a 63-character set system, the build procedure NIP5870 must be rerun. The job requires Pascal. If you did not order Pascal, you can install a 63-character set version by entering this command:

```
X.SYSGEN(SWAP,density,NIP)
```

Replace density with the tape density of the deadstart tape.

For more information about SYSGEN(SWAP), refer to chapter 8.

Protocol File Transfer Facility (PFTF)

To obtain a 63-character set version of PFTF, enter the following command:

```
X.SYSGEN(PFTF,C63)
```

This command replaces the two files RMUGET and PFTFTR on user name LIBRARY.

The Remote Diagnostic Facility (RDF) allows use of an interactive terminal when networks are not yet installed. Thus, you can use RDF as an interface for using a NOS editor to create your deadstart deck files or configuration files. For more information on how to use RDF, see the NOS Online Maintenance Software Reference Manual.

To initiate RDF, follow these steps:

1. When you deadstart 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 Version 2 Analysis Handbook for the syntax of the entry.)
2. After you deadstart 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 is the EST ordinal of the two-port multiplexer.

3. Enable the RDF subsystem by entering these commands:

```
SUBSYST.  
L.ENABLE,RDF.  
L.END.
```

4. 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 type-ahead feature is not available on RDF.
- Full Screen Editor (FSE) does not function properly.

To minimize installation time and confusion and to establish conventions that enable better CDC service/support to users, the preferred configurations are defined in table E-1.

Complete configurations are not defined in table E-1 due to differing customer requirements. Conventions are specified for a base system. Systems configured according to the conventions provide these 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 E-1. Preferred Configuration

Channel	CYBER 170/180 Models 815, 825, 835, 840(A), 845, 850(A), 855 860(A), 865, 870(A), 875, 960, 990, 994, 995	CYBER 180 Models 810, 830 Group A	CYBER 180 Models 810, 830 Group B	CYBER 180 Models 810, 830 Group C
0	+	First ISD RMS	First ISD RMS	First ISD RMS
1	First RMS	+	First non-ISD RMS	Third ISD RMS
2	Second RMS	-	-	-
3	Third RMS	+	Second non-ISD RMS	Fourth ISD RMS
4	Fourth RMS	+	Third non-ISD RMS	+
5	First Communications	-	-	-
6	Second Communications	First ISMT MT	First ISMT MT	First ISMT MT
7	Third Communications	Communications	First Communications	Communications
10	Console	18002-2 Console	18002-2 Console	18002-2 Console
11	First MT	+	+	+
12	Unit Record	+	Unit Record	+
13	+	-	-	-
20	+	Second ISD RMS	Second ISD RMS	Second ISD RMS
21	Fifth RMS	+	Fourth RMS	+
22	Sixth RMS	-	-	-
23	Seventh RMS	+	+	+
24	Eighth RMS	+	+	+
25	+	-	-	-

(Continued on next page)

Table E-1. Preferred Configuration

	CYBER 170/180 Models 815, 825, 835, 840(A), 845, 850(A), 855 860(A), 865, 870(A), 875, 960, 990, 994, 995	CYBER 180 Models 810, 830 Group A	CYBER 180 Models 810, 830 Group B	CYBER 180 Models 810, 830 Group C
--	--	--	--	--

(Continued from previous page)

26	+	ISD or ISMT	ISD or ISMT	ISD or ISMT
27	+	+	+	+
30	+	-	-	-
31	Third MT	+	First non-ISMT MT	+
32	Fourth MT	+	Second non-ISMT MT	+
33	Second MT	-	-	-

Group A specifies channel assignments for an 810/830 using only Intelligent Small Disk (ISD) (834s and 836s) and Intelligent Small Magnetic Tape (ISMT) (639s) peripherals.

Group B specifies channel assignments for an 810/830 with both ISD/ISMT peripherals and other CDC disk and tape peripherals.

Group C specifies channel assignments for an 810/830 with the 18002-2 console option.

The + symbol means the channel is available for CYBER channel peripheral connection.

The - symbol means the channel is not available for CYBER channel peripheral connection.

Index

- 5870 installation C-2
- 63-character set, installation C-1

- A**
- AAM2 - CYBER Record Manager
Advanced Access Methods Version 2
 - Additional procedures 7-4
 - USER file directives 7-4
- APL2 - APL Version 2
 - APL entry message 7-7
 - APL validation requirements 7-8
 - SYSGEN functions 7-7
 - Unique parameters 7-7
- ATF - Automated Tape Facility
 - ATF command 7-9
 - File placement 7-9
 - Unique parameters 7-9
- Audience 7
- Automatic file installation, override 3-7, 4-8

- B**
- BAM - CYBER Record Manager Basic
Access Methods Version 1
 - Installation parameters 7-10
 - Installation procedure messages 7-10
- BASIC3 - Basic Version 3
 - Installation parameters 7-11
 - Installation procedure messages 7-11

- C**
- CCL - CYBER Control Language Version
1
 - Additional procedures 7-14
 - Installation parameters 7-12
- CCP - CYBER CROSS System and
Communications Control Program
 - Binary installation option 7-23
 - Build steps description 7-19
 - CCP extra PLs definition 7-42
 - CCP load file definition 7-42
 - CCP system definition 7-38
 - CCP variant load module definition 7-40
 - CCP/CROSS installation examples 7-43
 - CCP/CROSS permanent files 7-26
 - CCPBLB - CCP binary library 7-34
 - CCPEDIT - Edit variant module 7-36
 - CCPLOAD - Generate CCP load file 7-37
 - CCPPH1 - CCP Phase 1 7-32
 - CCPPURG - CCP/CROSS installation
files purge 7-38
 - CCPVAR - CCP variant 7-35
 - Configuration information 7-17
 - CROSS - CROSS system installation 7-31
 - General build step call 7-24
 - Hardware requirements 7-16
 - Installation parameters for CROSS 7-32
 - Released software variants 7-17
 - Security character parameters 7-30
 - USERBPS file 7-29
- CDCNET - Control Data Distributed
Communications Network
 - Activating new level 7-62
 - CDCNET Host Application (CHA1)
product content 7-65
 - CDCNET Network Host Application
program requirements 7-70
 - Configuration information 7-56
 - INITMDI - Initialize MDI 7-74
 - Installation parameters 7-66
 - Installation wrapup activities 7-64
 - Installing first time 7-57
 - Installing for MDI reset support only 7-63
 - Installing separate TIP 7-62
 - Message templates 7-68
 - NAM startup procedure file changes 7-69
 - NETFS - Network File Server 7-71
 - NETLS - Network Log Server 7-72
 - NETOU - Network Operator Utility 7-73
 - Network Definition Language (NDL)
requirements 7-76
 - NPA database maintenance 7-64
 - Unique parameters 7-66
 - Upgrading 7-59
 - USER file directives 7-66
- CDCNET batch critical update (BCU)
 - Cleanup 5-7
 - Preliminaries 5-5

Procedure	5-5	Using PFLOAD	9-10
CDCS2 - CYBER Database Control		With Auxiliary Packs	9-11
System Version 2		Documentation	
Accounting table	7-77	Audience	7
CDC Procedure file	7-77	Conventions	9
Special notes	7-77	Customer support hotline	10
Unique parameters	7-77	Organization	8
CID - CYBER Interactive Debug Version		Related publications	11
1	7-80	Submitting comments	10
CML - Concurrent Maintenance Library		DUAL - Dual State	
	7-80	Configuration information	7-85
COBOL5 and COBOL5Q - COBOL		Dual state on older NOS systems	7-90
Version 5		Dual state source library	7-86
Installation parameters	7-82	Dual state upgrade of NOS only	7-87
Installation procedures	7-81	Minimum hardware requirements	7-92
Unique parameters	7-81	NOS ACCFILE	7-96
COMMOD file parameters	9-1	NOS ACCFILE modification	7-98
Configuration file requirements		NOS and NOS/VE memory allocation	7-90
(preparation)	2-4	NOS RUNJOBS modification	7-95
Configure the system		NOS SETVE modification	7-94
Configuration files	2-14	NOS/VE initiation	7-93
Deadstart decks	2-14	Peripheral processor allocation	7-91
		Unique parameters	7-84
D		VEIAF family name selection	7-93
DCL - Data Catalogue 2 Version 2.0		Dual state batch critical update (BCU)	
Product dependencies	7-83	Correcting dual state partner	5-8
SYSGEN functions	7-83	Correcting earlier version of NOS	5-9
Deadstart deck requirements		Correcting older version of NOS/VE	5-10
(preparation)	2-3	F	
Deadstart new system		FDBF - FORTRAN Data Base Facility	
Component order installation	4-11	Version 1	
First time installation	2-15	Product dependencies	7-99
Upgrade installation	3-13	Unique parameters	7-99
Deadstart system	2-9	File conversions	C-2
Deadstart tape binaries, customizing		File formats	
Build products from source code	6-9	Order of products	B-1
Create files GLOBLIB and PRODUCT		Permanent File Tapes	B-3
	6-6	FSE - Full Screen Editor	
Create new deadstart tape	6-22	FSEEX and SMFEX implementations	
Create USER files	6-4		7-100
Execute MODOPL procedure	6-8	Installation Parameters	7-102
Installation procedures	6-3	Site-defined teach file	7-102
Load installation tools and files	6-2	SMF procedure file	7-101
Run Group 1 through 6	6-12	SMFSTAT Procedure	7-103
USERF parameter	6-5	SYSGEN functions for FSE	7-102
Deadstart tape, writing	3-6, 4-7		
DECKLIS procedure	9-9		
Disk installations			

FTN4, FTNTS, FTN5	
Installation parameters for FTN4 and FTNTS	7-104
Installation parameters for FTN5	7-104
MODEL parameter	7-104
FTP	7-198

G

GENDST procedure	9-14
----------------------------	------

H

Hardware inventory (preparation)	2-2
--	-----

I

IAF - Interactive Facility Version 1	
IAF procedure file	7-105
Installation parameters	7-105
Special notes	7-106
Initial deadstart (preparation)	2-5
Installation	
5870	C-2
63-character set	C-1
Commands, parameters, and procedures	9-1
Component order	4-1
Concurrent Maintenance Library (CML)	2-16
Corrective code	5-1
Customizing	6-1, 1-1
Definition	1-1
Dual state	1-2
First time	2-1
Online manuals	2-16
Permanent files	3-12, 4-10, 2-11
Preparing for	1-3
Special product information	7-1
Tools, set up	3-3, 4-2
Types	1-1
Upgrade	3-1
Installation tools, set up	
Permanent file based products	4-4
Products requiring configuration tools	4-3
Standard products	4-5
ITF - Interactive Transfer Facility	
Version 1	
ITF command	7-108

Product dependencies	7-107
--------------------------------	-------

L

LCS3 and FCS3 - Conversion Aids	
System Version 3	
USER file directives for LCS3	7-109
LOADER - CYBER LOADER Version 1	
Installation parameters	7-112
Unique parameters	7-111

M

MAP Subsystem	
MAP installation procedures	7-113
MSSI validation requirements	7-113
Procedures for initiating MSSI	7-114
MCS - Message Control System Version 1	
Installation parameters	7-115
MCS procedure file	7-116
Special notes	7-117
Unique parameters	7-115
MISCGET procedure	9-15
MSE - Mass Storage Extended Subsystem	
COMBFAS parameters	7-119
Common deck parameters	7-119
COMXIPR parameters	7-119
Configuration information	7-118
Unique parameters	7-118

N

NAM/CCP (255x requirements and information)	2-13
NAM/CDCNET (DI requirements and information)	2-12
NAM5/NAM5D - Network Access Method	
Version 1	
Configuration information	7-120
CS - Communications Supervisor	7-136
JOB directive	7-127
Job skeleton records	7-129
NAM procedure file	7-124
Network Host Product (NHP) program requirements	7-135
Network startup master file	7-126
NIP - Network Interface Program	7-137
NS - Network Supervisor	7-140
NVF - Network Validation Facility	7-141

PARAM directive	7-126
Parameter records	7-126
Site-defined user names	7-122
Special notes	7-123
TITLE statement	7-126
Unique parameters	7-124
USER file directives	7-125
Network activation	2-13
Network terminal, bringing up	2-12
NIP5870 - 5870 Printer	
Installation procedure messages	7-142
NOS and NOS2B - Network Operating System	
819 PPU driver installation	7-163
CALLxxx decks	7-144
COMSACC parameters	7-146
COMSBIO parameters	7-147
COMSIOQ parameters	7-147
COMSJIO parameters	7-148
COMSLSD parameters	7-148
COMSMLS parameters	7-149
COMSMMF parameters	7-149
COMSMSC parameters	7-150
COMSMTX parameters	7-151
COMSPFM parameters	7-154
COMSPRO parameters	7-157
COMSREM parameters	7-157
COMSRSX parameters	7-158
COMSSCD parameters	7-159
COMSSFS parameters	7-159
COMSSRU parameters	7-159
COMSSSJ parameters	7-159
COMSVED parameters	7-160
COMTNAP parameters	7-160
Configuration information	7-143
DSD parameters	7-163
Installation parameters	7-144
PPCOM parameters	7-162
NSS - NOS Scope 2 Station Facility 1	
Buffer size parameters	7-170
Configuration information	7-164
Installation parameters	7-168
Installation procedure messages	7-165
Spooling and recalling time parameters	7-169
Spun-Off Task (SPOT) memory requirements	7-166
Table size parameters	7-168

O

Online manuals, installation	2-16
--	------

P

Passwords	2-16
Permanent files	
Customizing	6-23
Installation	3-12, 4-10, 2-11
Protocol File Transfer Facility (PFTF)	C-2
PSU - Printer Support Utility Version 1.1	
Configuration information	7-171
File placement	7-173
NDL file entries	7-174
PSU command	7-173
Unique parameters	7-173

Q

QU3 - Query Update Version 3	7-174
--	-------

R

RBF5/RBF5D - Remote Batch Facility	
Version 1	
File placement	7-176
RBF command	7-175
Special notes	7-175
Unique parameters	7-175
USER file directives	7-176
RDFEX - Remote Diagnostic Facility	7-177
Remote Diagnostic Facility (RDF)	D-1
REPORT procedure	9-15
RESETP procedure	9-16
RHF - Remote Host Facility Version 1	
Configuration information	7-178
Hardware configuration	7-180
Installation parameters	7-180
NAD microcode	7-181
NAD microcode initialization parameters	7-181
RHF procedure file	7-179
RHP - PTF/QTF File Transfer Facility	
Configuration information	7-182
Installation parameters	7-185
QTF configuration directives	7-186
QTF initiation procedure parameters	7-186

Unique parameters	7-184
S	
SETUP procedure	9-17
Software inventory (preparation)	2-3
Software support hotline	10
SOLVER modification sets	
Adding corrective code	5-2, 5-3
Creating new permanent OPL	5-2
Creating permanent NEWPL	5-4
Putting code on USER file	5-2, 5-4
Using modify for NOS and NOS products	5-1
Using update for common product set, network host products, CCP, and CROSS	5-3
SORT5 - Sort/Merge Version 5	
Installation procedure messages	7-186
Subsystem initiation	
Automatic	7-1
Enable and Disable commands	7-1
Subsystem startup files	7-1
SYSGEN	8-1
Calling	8-1
Loading permanent files	8-2
Loading RECLAIM database	8-2
SYSGEN Functions	
SYSGEN(COPYSYS)	8-5
SYSGEN(COPYTAP)	8-6
SYSGEN(DST)	8-7
SYSGEN(MOVE)	8-7
SYSGEN(SWAP)	8-8
SYSGEN(UPGRADE)	8-9
SYSGEN maintenance	8-9
SYSGEN validations	8-10
SYSGEN(FILES)	8-4
SYSGEN(FULL)	8-4
SYSGEN(group) functions	8-4
SYSGEN(RECLAIM)	8-5
SYSGEN(SOURCE)	8-5
SYSGEN(xxxx) functions	8-3
System configurations, preferred	E-1

T

TAF - Transaction Facility Version 1	
Installation parameters	7-189
TAF procedure file	7-188
TAF validation requirements	7-188
Task library	7-188
Unique parameters	7-187
TCP/IP	7-198
TCP/IP/FTP/TELNET	
FTP Network Host Application program requirements	7-200
FTPI - File Transfer Protocol Interpreter/Server	7-201
FTPS - File Transfer Protocol Server	7-201
Host Name and address definition for TCP/IP network	7-198
NAM startup procedure file changes	7-200
Network Definition Language (NDL) requirements	7-202
Unique parameters	7-198
TELNET	7-198
TERMLIB - Terminal Library	
Unique parameters	7-204
TEXT and TEXTIO - Product TEXTS and Product TEXTS I/O	
Default IPARAMS	7-205
Installation parameters	7-207
Installation procedure messages	7-211
MFT parameter details	7-208
Unique parameters	7-206

U

UPDATE	7-212
Update decks and configuration files	
Creating or updating configuration files	3-5, 4-6
Updating deadstart decks	3-5, 4-6
User names	2-16
User names, creation	3-2, 4-2

