60459320

CONTROL DATA

NOS VERSION 2 INSTALLATION HANDBOOK

CDC® COMPUTER SYSTEMS:

CYBER 180

CYBER 170

CYBER 70

MODELS 71, 72, 73, 74

6000

REVISION RECORD

REVISION	DESCRIPTION
A (04-26-82)	Handbook released; reflects NOS 2.0 (which succeeds NOS 1.4) at PSR level 562/552 and supersedes the NOS Version 1 Installation Handbook. It documents the CYBER 170 Computer System, Models 825, 835, and 855, and 819 disks; includes a section on APRDECK entries; adds IOB, PPB, DOWN, ISHARE, and PRESET CMRDECK entries; drops documentation of ACCOUNT, AUTOROLL, BATCHIO, EI200, MAGNET, MSS MASTER, PRIORITY, QPROTECT, TELEX, USERS, and VALID IPRDECK entries; adds BIO, EXTENDED STACK PURGING, LOGGING, MAP, MASTER MSS, SECONDARY USER CARDS, and UNLOCK IPRDECK entries; and documents the changed format of IPRDECK entries. The handbook also documents the two new common disk area utility options (environment interface and microcode) and three new decks for NOS installation parameters (COMSMSC, COMSSFS, and COMSSRU). It drops documentation of decks COMPCMX, COMSEXP, and RESEQ, CPUMTR segmentation, and procedure file LIBMOD. The handbook drops documentation of Advanced Access Methods Version 1, ALGOL Version 4, COBOL Version 4, COBOL Version 4 to COBOL Version 5 Conversion Aids, CYBER Database Control System Version 1, Data Description Language Version 2, Time-Sharing Module, Export/Import Module, and TAF/TS Transaction Facility Data Manager. The handbook also drops documentation of the following equipment: 841, 65x, 6671/6676/2550-100.
B (01-27-83)	Revised to reflect NOS 2.1 at PSR summary level 580/577. This revision documents the CYBER 170 Computer System models 815, 865, and 875, and the 885-42 Disk Storage Subsystem; adds the Remote Diagnostic Facility and Remote Host Facility; revises installation procedures for NOS 2, Cross, CCP, and NAM; drops the use of procedures GENJOB and SYSTEM; adds procedures SETUP and MISCGET; adds production environment installation through use of a global library; adds LID, PRIVILEGED RDF, and RESIDENT RDF IPRDECK entries; adds short printer paper parameter to unit record EST entry; adds two-port multiplexer, network access device, and MAP III equipment EST entries; adds EST entry for the 885-42 disk; drops IOB and UEC CMRDECK entries; adds LIDT and XM CMRDECK entries; revises release tape contents; and includes editorial and technical corrections. Because of extensive changes, change bars and dots are not used. This edition obsoletes all previous editions.
C (10-11-83)	Revised to reflect NOS 2.2 at PSR summary level 596/587. This revision documents the installation procedures for the Full Screen Editor, Interactive Transfer Facility, and NOS-SCOPE 2 Station Facility. It documents changes to the CMRDECK and addition of the EQPDECK. This revision adds the following items for multilevel security: deck COMSMLS, CMRDECK entry OPSECM, EQPDECK entry ACCESS, and IPRDECK entries SECCATS, OQSH, and SECURES. It adds parameters for secure login; adds information for the multihost capability; describes IPRDECK entries COMLIB, EI, MICRO, MEMORY CLEARING, SPC, SPD, SPL, SPW, CLASS, and PCLASS; describes the modified MOVEPF utility; adds verification for RHF; documents CTI changes; and adds more description of DECKOPL parameters and procedures. This edition obsoletes all previous editions.
D (10-12-84)	Revised to reflect NOS 2.3 at PSR level 617/617. This revision documents the Tailored Release Process (TRP). Deadstart deck and CIP documentation have been deleted and moved to the Analysis Handbook and the CIP User's Handbook. Because of extensive changes, change bars and dots are not used and all pages reflect the current revision level. This edition obsoletes all previous editions.
E (03-22-85)	Revised to reflect NOS 2.4.1 at PSR level 630. This revision documents the CYBER 180 Computer Systems Models 840, 850, and 860, the 895 Disk Storage Subsystem, and installation information for Concurrent Maintenance Library (CML) and Mass Storage Archival Subsystem (MAS). This revision obsoletes all previous editions.
F (09-30-85)	Revised to reflect NOS 2.4.2 at PSR level 642. This revision documents modifications to DECKOPL procedures, gives more detailed documentation for controlware installation, and documents the CYBER 180 Computer System Model 990. This revision obsoletes all previous editions.
Publication No. 60459320	

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

Address comments concerning this manual to:

Control Data Corporation
Technology and Publications Division
4201 North Lexington Avenue
St. Paul, Minnesota 55126-6198

or use Comment Sheet in the back of this manual.

REVISION RECORD

REVISION	DESCRIPTION
G 12/16/85	Revised to reflect NOS 2.4.3 at PSR level 647. This revision documents installation of CDCNET, gives information on the upgrade process from one PSR level to the next, and adds additional enhancements to the SYSGEN documentation.
H 09/25/86	Revised to reflect NOS 2.5.1 at PSR level 664/650. This version gives specific instructions for performing four types of installations: first-time, upgrade, component, and corrective. It also gives instructions for installing dual-state systems and CDCNET. Because of extensive changes, change bars and dots are not used and all pages reflect the current revision level. This edition obsoletes all previous editions.

Publication No.
60459320

LIST OF EFFECTIVE PAGES

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

PAGE	REV
Front Cover	-
Title Page	-
2	H
3	H
4	H
5	H
6	H
7	H
8	H
9	H
10	H
11	H
12	H
13/14	H
15	H
16	H
17	H
18	H
19	H
Divider	-
1-1	H
1-2	H
1-3	H
1-4	H
Divider	-
2-1	H
2-2	H
2-3	H
2-4	H
2-5	H
2-6	H
2-7	H
2-8	H
2-9	H
2-10	H
2-11	H
2-12	H
2-13	H
2-14	H
2-15	H
2-16	H
Divider	-
3-1	H
3-2	H
3-3	H
3-4	H
3-5	H
3-6	H
3-7	H
3-8	H
3-9	H
3-10	H
3-11	H
3-12	H
Divider	-
4-1	H
4-2	H
4-3	H
4-4	H
4-5	H
4-6	H
4-7	H
4-8	H
Divider	-
5-1	H
5-2	H
5-3	H
5-4	H
5-5	H
5-6	H
5-7	H
5-8	H
Divider	-
6-1	H
6-2	H
6-3	H
6-4	H
6-5	H
6-6	H
6-7	H
6-8	H
6-9	H
6-10	H
6-11	H
6-12	H
6-13	H
6-14	H
6-15	H
6-16	H
6-17	H
6-18	H
6-19	H
6-20	H
Divider	-
7-1	H
7-2	H
7-3	H
7-4	H
7-5	H
7-6	H
7-7	H
7-8	H
7-9	H
7-10	H
7-11	H
7-12	H
7-13	H
7-14	H
7-15	H
7-16	H
7-17	H
7-18	H
7-19	H
7-20	H
7-21	H
7-22	H
7-23	H
7-24	H
7-25	H
7-26	H
7-27	H
7-28	H
7-29	H
7-30	H
7-31	H
7-32	H
7-33	H
7-34	H
7-35	H
7-36	H
7-37	H
7-38	H
7-39	H
7-40	H
7-41	H
7-42	H
7-43	H
7-44	H
7-45	H
7-46	H
7-47	H
7-48	H
7-49	H
7-50	H
7-51	H
7-52	H
7-53	H
7-54	H
7-55	H
7-56	H
7-57	H
7-58	H
7-59	H
7-60	H
7-61	H
7-62	H
7-63	H
7-64	H
7-65	H
7-66	H
7-67	H
7-68	H
7-69	H
7-70	H
7-71	H
7-72	H
7-73	H
7-74	H
7-75	H
7-76	H
7-77	H
7-78	H
7-79	H
7-80	H
7-81	H
7-82	H
7-83	H
7-84	H
7-85	H
7-86	H
7-87	H
7-88	H
7-89	H
7-90	H
7-91	H
7-92	H
7-93	H
7-94	H
7-95	H
7-96	H
7-97	H
7-98	H
7-99	H
7-100	H
7-101	H
7-102	H
7-103	H
7-104	H
7-105	H
7-106	H
7-107	H
7-108	H
7-109	H
7-110	H
7-111	H
7-112	H
7-113	H
7-114	H
7-115	H
7-116	H
7-117	H
7-118	H
7-119	H
7-120	H
7-121	H
7-122	H
7-123	H
7-124	H
7-125	H
7-126	H
7-127	H
7-128	H
7-129	H
7-130	H
7-131	H
7-132	H
7-133	H
7-134	H
7-135	H
7-136	H
7-137	H
7-138	H
7-139	H
7-140	H
7-141	H
7-142	H
7-143	H
7-144	H
7-145	H
7-146	H
7-147	H
7-148	H
7-149	H
7-150	H
7-151	H
7-152	H
7-153	H
7-154	H
7-155	H
7-156	H
7-157	H
7-158	H
7-159	H
7-160	H
7-161	H
7-162	H
7-163	H
7-164	H
7-165	H
7-166	H
7-167	H
7-168	H
7-169	H
7-170	H
7-171	H
7-172	H
7-173	H
7-174	H
7-175	H
7-176	H
7-177	H
7-178	H
7-179	H
7-180	H
7-181	H
7-182	H
7-183	H
7-184	H
7-185	H
7-186	H
7-187	H
7-188	H
7-189	H
Divider	-
8-1	H
8-2	H
8-3	H
8-4	H
8-5	H
8-6	H
8-7	H
8-8	H
8-9	H
8-10	H
8-11	H

PAGE	REV
Divider	-
9-1	H
9-2	H
9-3	H
9-4	H
9-5	H
9-6	H
9-7	H
9-8	H
9-9	H
9-10	H
9-11	H
9-12	H
9-13	H
9-14	H
9-15	H
9-16	H
Divider	-
A-1	H
A-2	H
A-3	H
A-4	H
A-5	H
A-6	H
A-7	H
A-8	H
A-9	H
A-10	H
A-11	H
A-12	H
A-13	H
A-14	H
A-15	H
A-16	H
Divider	-
B-1	H
B-2	H
B-3	H
B-4	H
B-5	H
B-6	H
B-7	H
B-8	H
B-9	H
B-10	H
B-11	H
B-12	H
Divider	-
C-1	H
C-2	H
C-3	H
Divider	-
D-1	H
Divider	-
E-1	H
E-2	H
Divider	-
F-1	H
F-2	H
F-3	H
F-4	H
F-5	H
F-6	H
F-7	H
F-8	H
F-9	H
F-10	H
Divider	-
Index-1	H
Index-2	H
Index-3	H
Index-4	H
Index-5	H
Index-6	H
Comment Sheet	H
Back Cover	-

PREFACE

PURPOSE

This handbook describes the installation of the CONTROL DATA® Network Operating System (NOS) Version 2.5.1 and its products. NOS controls the operation of the following computer systems:

- CDC® CYBER 180 Computer Systems
  Models 810, 830, 835, 840, 845, 850, 855, 860, and 990

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

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

- CDC 6000 Computer Systems

AUDIENCE

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

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

MANUAL ORGANIZATION

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

This handbook is organized in the following manner:

- Chapter 1 briefly 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 contains special information for configuring and customizing NOS and each of its products.

- Chapter 8 describes the SYSGEN utility.

- Chapter 9 contain descriptions of special installation commands, parameters, and procedures.

- The six appendixes contain reference information: a glossary of terms, the file formats of the source materials, special information for a 63-character set installation, Remote Diagnostic Facility information, system configurations, and installation verification information.

CONVENTIONS

The following conventions are used throughout this handbook:

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

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

- The terms command and control statement are interchangeable.

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

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

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

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

- The CDC 18002-2 console is available as an option for CYBER 180 models 810 and 830 systems using the NOS 2.3 level 617 or a later operating system. This product includes a CDC 634B display terminal (also known as the 721-21 display terminal) and an AV117A cable. This console is referred to throughout this handbook as CC634B.

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

- 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.

SUBMITTING COMMENTS

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

Control Data Corporation
Technology and Publications Division
ARH219
4201 Lexington Avenue North
St. Paul, MN 55126-6198

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

RELATED PUBLICATIONS

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

The NOS 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 can be found in the COMPASS Reference Manual and in the appropriate computer system hardware reference manual.

EXTENDED MEMORY MANUALS

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

NOS 2 MANUALS

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

OPTIONAL PRODUCT MANUALS

Control Data Publication	Publication Number
APL Version 2 Reference Manual	60454000
BASIC Version 3 Reference Manual	19983900
CDCNET Configuration and Site Administration Guide	60461550
CDCNET Network Operations Manual	60461520
COBOL Version 5 Reference Manual	60497100
Concurrent Maintenance Library Reference Manual	60455980
Conversion Aids System Version 3 Reference Manual	19265358
CYBER Cross System Version 1 Build Utilities Reference Manual	60471200
CYBER Cross System Version 1 Macro Assembler Reference Manual	96836500
CYBER Cross System Version 1 Micro Assembler Reference Manual	96836400
CYBER Cross System Version 1 PASCAL Reference Manual	96836100
CYBER Interactive Debug Version 1 Reference Manual	60481400
Data Catalogue 2 Version 2 Reference Manual	60483350
DMS-170 CYBER Database Control System Version 2 Application Programming Reference Manual	60485300
DMS-170 CYBER Database Control System Version 2 Data Administration Reference Manual	60485200
DMS-170 Query Update/CYBER Record Manager Data Administration Reference Manual	60482100
FORTRAN Data Base Facility Version 1 Reference Manual	60482200
FORTRAN Extended Version 4 Common Library Mathematical Routines Reference Manual	60498200
FORTRAN Extended Version 4 Reference Manual	60497800
FORTRAN Extended Version 4 to FORTRAN Version 5 Conversion Aids Program Version 1 Reference Manual	60483000
FORTRAN Version 5 Common Library Mathematical Routines Reference Manual	60483100
FORTRAN Version 5 Reference Manual	60481300
MSSI Version 3 Installation Handbook	60458830
MSSI Version 3 Reference Manual	60458820

Control Data Publication	Publication Number
Message Control System Version 1 Reference Manual	60480300
Network Access Method Version 1 Host Application Programming Reference Manual	60499500
Network Access Method Version 1/Communications Control Program Version 3 Terminal Interfaces Reference Manual	60480600
Network Definition Language Version 1 Reference Manual	60480000
NOS Full Screen Editor User's Guide	60460420
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
Screen Formatting Reference Manual	60460430
STIMULA 1 Reference Manual	60234800
Sort/Merge Version 5 Reference Manual	60484800
TAF Version 1 Reference Manual	60459500
TAF/CRM Data Manager Version 1 Reference Manual	60459510
Text Editor Version 1 Reference Manual	60436100
Update Version 1 Reference Manual	60449900
XEDIT Version 3 Reference Manual	60455730
8-Bit Subroutines Version 1 Reference Manual	60495500

HARDWARE MANUALS

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

DISCLAIMER

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

CONTENTS

APPENDIXES

INDEX

FIGURES

TABLES

CHAPTER 1

INTRODUCTION

INTRODUCTION

Installation is the process of creating an operating system deadstart file and then using the deadstart file to load the operating system and products onto 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

There are four types of NOS system installations 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 begun from the system console and includes instructions 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 level of the components you want to install must be the same as that of the NOS software running on your system.

- Corrective Code and BCU Installations. 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 will be performing, you can customize NOS or a product during installation. If you want to customize portions of the software you are installing, 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 to the NOS/VE Installation Manual.

If you want to	Then
Perform a first-time installation of NOS and NOS/VE	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 Manual and follow the instructions for a first-time installation of dual state.
Add dual state and upgrade your NOS system at the same time	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 Manual and follow the instructions for a first-time installation of dual state.
Add dual state and not update your current NOS system	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 Manual and follow the instructions for a first-time installation of dual state.
Upgrade dual state and NOS	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 Manual and follow the instructions for upgrading a dual-state system.

If you want to	Then
Upgrade only dual state and not upgrade NOS	1. Follow the instructions in this handbook for an upgrade installation. However, skip the steps that describe how to update NOS products. 2. Once you have completed the NOS installation and have re-deadstarted your system, go to the NOS/VE Installation Manual and follow the instructions for upgrading a dual-state system.
Apply corrective code to a dual-state system	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 Manual.

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 any 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 file copy of the SRB is released on the permanent file tapes.

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

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

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

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

CHAPTER 2

FIRST-TIME INSTALLATION

FIRST-TIME INSTALLATION 2

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 begun 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 and the steps are described in detail in the remaining sections of this chapter:

1. Prepare for Your 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. Wrapup

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 continue to the dual state chapter of the NOS/VE Installation Manual.

STEP 1: PREPARE FOR YOUR 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, you should 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, you should determine:

- Peripheral type (for example, 895 disk)
- Channel or channels 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 that depicts 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 you have 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 will be 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) which 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 2 Analysis Handbook for more information about deadstart deck entries for both hardware and software. Chapter 7 of this handbook contains 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 for creating deadstart decks for your site. You can also use the preconfigured decks to perform the first deadstart of the system before installing and configuring the system. Table 2-3 lists the released deadstart decks; refer to Initial Deadstart Preparation 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	CMRDECK	EQPDECK	APRDECK
Dual State	Y	Y	
Mass Storage Extended		Y	
NAM/CCP Network		Y	
NAM/CDCNET Network		Y	
NOS	Y	Y	Y
NOS Scope Station Facility		Y	
PTF/QTF Transfer Facility	Y		
Remote Host Facility	Y	Y	

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 the table.

Table 2-2. Software Products and Configuration Files Required

Product	RCFMid	LIDCMid	LIDLIST	CDCNET†	NCFFILE	LCFFILE	NLFFILE	MSECONF	EVFULFN
Dual State		X	X						
Mass Storage Extended								X	
NAM/CCP Network					X	X	X		
NAM/CDCNET Network				X		X			
NOS Scope Station Facility		X							
Printer Support Utility				X	X	X			X
PTF/QTF (NAM/CCP)		X			X	X	X		
PTF/QTF (NAM/CDCNET)		X		X		X			
PTF/QTF (RHF)	X	X							
Remote Host Facility	X	X							

†CDCNET has numerous permanent 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) use megabytes.

- CMRD00 is a directory for all released deadstart decks.
- CMRD35 is an unconfigured CMRDECK, i.e., contains no equipment definitions.
- IPRDECK 01 is for the 810/830 which uses the asynchronous printer and 639 tape drives.
- IPRDECK 00 is for all other mainframes.

- LIBDECK 00 is for mainframes without UEM or ESM.
- LIBDECK 01 is for mainframes with ESM.
- LIBDECK 02 is for mainframes with UEM.

Select the decks that describe a configuration that closely matches 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 systems and temporary files 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 which you will assign to NOS/VE in your actual production system. Also, you do not need to define equipment for RHF, MSE, or the NOS Scope Station.

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

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

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	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
16	1000K		895	40,41
17	2000K		885-12	40,41
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	Total	Memory (Megabytes) NOS	UEM	VE	Disk Type
42	810/830	12	3.5	2	6.5	836
43	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 released separately from your NOS order and is sent to your customer engineer (CE). If you have an 800 series computer, the CIP modules must be installed to disk. If you have a non-800 series computer, installing the CIP modules is optional. Check with your CE to ensure that CIP has been installed; CIP installation should be a cooperative effort between you and your CE.

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 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-800s 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).

   OR

   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.

   OR

   If you are using a 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.

   OR

   For all other models, 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 or NOS/BE deadstart.

   c. Enter P to check the deadstart panel settings. Set the DISPLAY CMRDECK option to Y.

   d. Press the backspace key to return to the O display.

6. Advance to the CMRDECK display.

   For all CYBER 800 series computers and model 990, enter S to select 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.

   OR

   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, do one of the following:

      On the CC545 console, press the right blank key.

      OR

      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, do one of the following:

      On the CC545 console, press the right blank key.

      OR

      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 the SYSGEN command. SYSGEN will automatically create initial validation files for your system and load 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:

   *****END SYSGEN*****

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. Read the hardware requirements below for the type of network you have installed. Then, follow the steps for activating the network.

CDCNET NETWORK (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 TDI or MTI 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 for completing 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) to define the line. NETOU is described in the CDCNET Operations Handbook.

To allow the terminal to be used, you must inform the CDCNET software of the system identifiers (serial numbers) of your DI´s. Do this by performing the commands below:

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

```
X.DIS.
USER,NETADMN,NETADMN.
BEGIN,ADDDI,DCNPLIB,type,si,pfn.
```

Parameter	Description
type	type of DI (MDI or MTI).
si	Last 6 digits of the DI´s system identifier. The system identifier is a 12-digit hexadecimal number that can be found inside each DI. The last 4 digits are a checksum. Do not include the checksum in this parameter.
pfn	Permanent file name for the configuration file to be created by ADDDI. This file name must be unique.

When the ADDDI procedure executes, you may see messages similar to the following:

```
NETFM-DELETE FAILED...
NETFM-DELETE FAILED...
NETFM-ADDED...
```

If you configured an MDI in step 1, perform step 2; if you configured an MTI, skip to step 3.

2. Execute the ADDDI procedure call again to define your TDI. Specify the DI´s system identifier (si) and a unique file name (pfn):

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

3. Terminate the DIS package:

```
DROP.
```

To allow the terminal to be used, initiate the network by following the steps in Activating the Network.

CCP NETWORK (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 allow the terminal to be used, initiate the network by following the steps in Activating the Network.

ACTIVATING THE NETWORK

To activate the network, perform the following steps:

1. Ensure that 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 a 255x NPU or for a 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 to load the 255x or DI hardware. NAM will submit numerous other applications to aid in the configuration and loading of the network.

4. Log in to the system. For CDCNET networks, you must first create a connection to the default NOS system name NOS_SYSTEM. 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 user name INSTALL. You will need an unlabeled scratch tape for the new deadstart tape.

1. Get a copy of the deadstart decks (CMRDECK, EQPDECK, IPRDECK, LIBDECK, and APRDECK) you used in STEP 2. Use the following commands:

   COMMON,SYSTEM.
   GTR,SYSTEM,DSTDECK.$deckname_1$,$deckname_2$,...,$deckname_n$

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

2. Use an available editor (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 commands to merge the records on file DSTDECK with the contents of your current system:

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

   Replace dd 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 2 Analysis Handbook for information on the PFDUMP utility.

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

```
CHECKPOINT 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 contains the permanent files needed to run the system, do not initialize those devices.

STEP 7: WRAPUP

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 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. The bound version of the online manuals is used by the EXPLAIN command. If you want to modify any of the online manuals, you can use the MANLOAD procedure to install the source version of the manuals. Because the online manuals use a great amount of disk space, you should delete any manuals you do not need once installation is completed.

INSTALLATION OF CONCURRENT MAINTENANCE LIBRARY (CML)

The permanent file tapes contain the most recent version of CML at the time NOS released. It is automatically installed as file CML3 to user name CDCCE by the X.SYSGEN(FULL) command. 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.

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 Manual.

CHAPTER 3

UPGRADE INSTALLATION

UPGRADE INSTALLATION 3

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 and are described in detail 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 Configuration Files and Decks
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 and any installation bulletins for any 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 name necessary for running the product. By creating the user names now, you can install files to the user name before deadstarting the new system. To determine the new user names, refer to these sources:

- The Software Release Bulletin lists any user names that are new for this release.
- Table 8-1 lists all of the user names on which the system will store files and the products that use those user names.

In addition to creating the new user names, you must update your ZZSYSGU permanent file that SYSGEN uses for executing USER commands. This file is described more completely under SYSGEN Validations in chapter 8. Your user names must be updated from the system console using the MODVAL utility. Refer to the NOS 2 Administration Handbook for details about MODVAL.

STEP 2: SET UP INSTALLATION TOOLS

This step sets up the RECLAIM database and a global library of tools to help in recompiling 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 released deadstart tape; if you will be customizing, you will also need the released permanent file tapes. Perform the following steps:

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

2. Copy the 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 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.
   ```

   If you will be customizing NOS or its products, perform the steps in chapter 6 of this handbook which describe customization. When you have completed those steps, return to STEP 3: Update Configuration Files and Decks 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:

- A new RECLAIM database, RECLDB, containing information about all of the files on the released permanent file tape. Any previous copy of file RECLDB will be replaced.

- Files PRODUCT and GLOBLIB. GLOBLIB contains the programs and procedures useful for recompiling configuration files and installing any new products for the first time. PRODUCT contains the SYSGEN procedures that you use for installing NOS and its products.

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 NCFFILE and LCFFILE.
NETFM	Network File Manager. Used for accessing the CDCNET Network Directory file (NETDIR).
NPA	Network Performance Analyzer. Used for analyzing CDCNET performance. Before using NPA, you must install a 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 tape and installing them on the system.

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

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

STEP 3: UPDATE CONFIGURATION FILES AND DECKS

During this step you will:

- Update the deadstart decks
- Create or update configuration files

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 2-1 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, GTR the desired decks from the released deadstart tape. Add the updated decks to file DSTDECK. Save file DSTDECK on 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. If you are installing a product for the first time and the product requires configuration files 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 install new products. Refer to the NAM 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 or upgrading CDCNET, refer to the CDCNET section of chapter 7 for special installation information.

STEP 4: WRITE A NEW DEADSTART TAPE

To write a new deadstart tape, use the 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 these steps from an interactive terminal logged in to user name INSTALL.

1. If the 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 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 (which you created in STEP 3) a local file.

3. If you did not customize any binaries, place LIBEDIT directives on local file USERD to add your local deadstart decks to the deadstart tape. If you do not need any directives, use a 0 in place of the USERD in step 4.

   If you did customize 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.

   a. 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 tape density you want to use for the new deadstart tape. A tape request will be issued for an unlabeled tape with VSN=NDT. The tape may be assigned from the system console using the VSN,est,NDT command. 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.

   b. If you customized any deadstart binaries, create the new deadstart tape using this command:

      ```
      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 VSN,est,NDT command. 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:

- Created all new configuration files
- Updated all existing configuration files
- Customized any other released permanent files

The files that normally reside on the special system user names NETOPS, SYSTEMX, LIBRARY, and SUBFAM0 (these are user names you cannot log in to from an interactive terminal) should be stored on user name INSTALL. Files that reside on login user names should have temporary file names. That way, the files will not be used until the system is deadstarted with the new release software.

In STEP 6 you will perform a two-phase installation: first you will use SYSGEN to automatically install files from the permanent file tapes; then you will move the files you have created and/or modified to their proper user names and file names. To eliminate unnecessary processing in phase 1, disable the loading of files that you have modified.

To disable the loading of a file or files, make temporary modifications to the RECLAIM database so that it appears as if the file or files are not on the permanent file tape. That is, delete the file names from the database. Table 3-1 lists each product that is released with a permanent file, the user name where the file resides, and the corresponding PFG file name on the permanent file tape. Refer to table 3-1 to make a list of all the files you do not want loaded to the new system.

NOTE

If you have installed CDCNET, do not disable PFGCHA1. NAMSTRT has not yet been updated to include this product.

Table 3-1. Automatic File Installation (Sheet 1 of 2)

Product Name	User/File Name(s)	PFG Name
APL 2	APL0/† APL1/†	PFGAPL2
Communications Control Program (CCP) 3	NETADMN/NLFFILE	PFGCCPL
Concurrent Maintenance Library (CML)	CDCCE/CML3	PFGCML1
Control Data Distributed Communications Network (CDCNET)	NETADMN/†, NETOPS/NETDIR NETOPS/NAMSTRT†† INSTALL/†	PFGDCNS PFGCHA1, PFGCHA2
CYBER Database Control System 2	SYSTEMX/CDC	PFGCDCS
Data Catalogue 2 Version 2.0	LIBRARY/†	PFGDCL2
Dual State	SYSTEMX/LIDCMid,LIDLIST INSTALL/VEMEM	PFGDUAL
Full Screen Editor	LIBRARY/FSEHELP, FSTEACH, FSEPROC SYSTEMX/SMF INSTALL/SMFSTAT	PFGFSE1
Interactive Facility (IAF) 1	SYSTEMX/IAF, IAFTM, IAFTR	PFGIAF1
Interactive Transfer Facility (ITF)	NETOPS/NAMSTRT ††	PFGITF1
Mass Storage Extended (MSE)	SYSTEMX/MSE,MSESLAV SUBFAM0/MSECONF	PFGMSE1
Mass Storage Subsystem (MSS) 1	SYSTEMX/MSS	PFGMSS1
Message Control System (MCS) 1	SYSTEMX/MCS	PFGMCS1
MMCL V3, MAP Macro Control Library	SYSTEMX/MAPLIBE, MAPLIBC	PFGMMCL

†There are numerous files installed with this product. Typically, these files are handled as a group and are not listed here. Consult chapter 7 in this handbook for each product and the table in appendix B for more information.

††The NAMSTRT file on user name NETOPS is updated by multiple products, all of which are listed in this table with double daggers (††). The NAMSTRT file is created initially by NAM and is added to by other products.

Table 3-1. Automatic File Installation (Sheet 2 of 2)

Product Name	User/File Name(s)	PFG Name
MSSI V3, MAP III	SYSTEMX/MAPCMI MAPECS,MAPCH INSTALL/MSSIP,MJOBSPL CDCCE/ CDCCE2/	PFGMSSI PFGAP1I,PFGAP1L
Network Access Method (NAM) 1	SYSTEMX/NAM,NAMNOGO NETOPS/NAMSTRT †† NETADMN/NCFFILE LCFFILE, NDLDATA	PFGNAM5 PFGNDL1
NOS Online Manuals	MANUALS/†	PFGMAN1
NOS SCOPE Station Facility	SYSTEMX/SSF	PFGNSS1
NOS 2 Package	MANUALS/† LIBRARY/PFTFTR,RMUGET LIBRARY/TDUFILE,TERMLIB LIBRARY/HELPLIB SYSTEMX/RDF INSTALL/SRB	PFGONLM PFGPFTF PFGTLIB PFGNOSB PFGRDF1 PFGSRB1
Printer Support Utility (PSU)	NETOPS/NAMSTRT††, EVFULFN	PFGPSU1
PTF/QTF File Transfer Facility	NETOPS/NAMSTRT††	PFGRHP1
Remote Batch Facility (RBF) 1	NETOPS/NAMSTRT††	PFGRBF5
Transaction Facility (TAF) 1	SYSTEMX/TAF KB100DC/TASKLIB	PFGTAF1
XEDIT 3	LIBRARY/XEDITH	PFGXEDT

†There are numerous files installed with this product. Typically, these files are handled as a group and are thus not listed here. Consult chapter 7 in this handbook for each product and the table in appendix B for more information.

††The NAMSTRT file on user name NETOPS is updated by multiple products, all of which are listed in this table with double daggers (††). The NAMSTRT file is created initially by NAM and is added to by other products.

Once you have made a list of the files you do not want loaded to the new system, perform the following steps from a terminal under user name INSTALL:

1. Execute the RECLAIM utility by entering this command:

   ```
   RECLAIM.
   ```

   RECLAIM will respond 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 will prompt you for a list of file names:

   ```
   ENTER FILE NAMES.
   ?
   ```

3. Enter the desired file names:

   $\text{filename}_1,\text{filename}_2,\ldots,\text{filename}_n$

   Separate the file names with commas (,). If you cannot fit all the file names on one line, follow the last file name with a comma (,) and then 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 will respond with a report of the file names that have been deleted. Then, it will display the following prompt:

   ```
   ENTER DIRECTIVE.
   ?
   ```

4. Terminate the RECLAIM session by entering the following:

   ```
   END
   ```

   RECLAIM will respond with the following message:

   ```
   RECLAIM COMPLETE.
   ```

Here are some additional notes on using RECLAIM:

- You can list the names of all files that have been deleted 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:

  $\text{RECLAIM,Z./RESET,UN=NS2psrin,PF=*/filename}_1,\ldots,\ \text{filename}_n$

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 has been terminated.
- Ensure that all subsystem activity is terminated (except for the MAG and B10 subsystems).

If desired, perform a permanent file backup using the PFDUMP utility. You can use the backup tapes if you need to restore any files. Refer to the NOS 2 Analysis Handbook for information about the PFDUMP utility.

To install new permanent files, follow the steps below. Enter the commands from the system console.

1. Access the new SYSGEN procedures:

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

2. If the NOS system you are upgrading is a pre-level 664/650 system, perform the following instructions. If not, skip to step 3.

   For NOS 2.5.1 L664/650 and later levels of NOS, you must ensure that file ZZSYSGU is present on user name SYSTEMX. This file contains user names and passwords for SYSGEN. To obtain the file, perform these instructions:

   a. Load the released deadstart tape to a local file named SYSTEM:

```
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; replace density with the density of that tape.

   b. Enter this command:

```
SYSGEN,LOADUSE.
```

   The first permanent file tape is requested and file ZZSYSGU is saved on user name SYSTEMX.

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

3. Have your permanent file tapes available for mounting and note the VSN of the first permanent file tape. The VSN can be found 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.

4. Press the period (.) key to cause the SYSGEN procedure to begin execution.

   SYSGEN will load files from the permanent file tapes and install them to their proper user names on the system. Once SYSGEN has completed installing files from the permanent file tape, you may proceed to install your site-modified files.

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

   USER,SYSTEMX,password.

6. Move a file to its proper location from user name INSTALL using the following command from DIS:

   SYSGEN,MOVE,pfn,un,ct,ac,m.

Parameter	Description
pfn	Name of the permanent file to be moved. The file must reside on the installation user name INSTALL.
un	User name to receive files. Valid user names and passwords are listed in table 8-1.
ct	Specifies the permanent file category. Enter PU for public, PR for private, or S for semiprivate. The default is PR.
ac	Specifies the alternate user CATLIST option. Enter Y to allow alternate users to CATLIST the file or N to prevent alternate users from using CATLIST to list the file. The default is N.
m	Assigns read permission to user name INSTALL. Enter PERMIT to allow read permission. The default does not give read permission. PERMIT should only be used if ct=PR.

Repeat steps 5 and 6 as many times as necessary to move all files. Refer to table 3-1 to determine the proper user name for each file.

If you recompiled files NCFFILE and LCFFILE, follow the instructions for upgrading the files in the NAM 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 installed an upgrade to CDCNET, perform the instructions in Activating A New CDCNET in the CDCNET section of chapter 7.

STEP 7: DEADSTART THE NEW SYSTEM

Before deadstarting the new system, ensure 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 4

INTRODUCTION

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

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

A component order installation consists of the steps listed below, and these steps are described in detail 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

STEP 1: CREATE NEW USER NAMES

For each product you are installing, create the user names necessary for running 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 which lists all of the user names on which the system will store files and the products that use those user names.

STEP 2: SET UP INSTALLATION TOOLS

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

- If your component order includes a product or products from table 4-1, follow the instructions in Products Requiring Configuration Tools.

Table 4-1. Products that Require Configuration

Remote Host Facility (RHF)
Control Data Distributed Communications Network (CDCNET)

- If your component order is comprised of only the products listed in table 4-2, follow the instructions in Permanent File Based Products.

Table 4-2. Permanent File Based Products

Conversion Aids Subsystems 3
Communications Control Program (CCP) 3
Cross
Data Catalogue
Lip under CCP 3
NOS Online Manuals
3270 TIP for CCP

- If the products in your component order are not listed in either table 4-1 or 4-2, follow the instructions in Standard Products.

PRODUCTS REQUIRING CONFIGURATION TOOLS

If your component order includes either RHF or CDCNET, you must load configuration tools to global library 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 tape. 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.
```

If you need to customize, refer to chapter 6 for more information. Then, continue with STEP 3: Update Configuration Files and Decks.

PERMANENT FILE BASED PRODUCTS

If your component order consists only of products that do not add binaries to the deadstart tape (table 4-2 lists these products), 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,dd.
   ```

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

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

This procedure updates the RECLAIM database, RECLDB, on your installation user name. Once this command has completed, you can perform the instructions in chapter 6 for customizing the products you are installing. If you do not need to customize any of the products, go to STEP 5: Override Automatic File Installation.

STANDARD PRODUCTS

If your component order does not contain any of the products listed in tables 4-1 and 4-2, 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 tape. 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.
```

If you need to customize, refer to chapter 6 for more information. Then, continue with STEP 3: Update Configuration Files and Decks.

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.
- 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.

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 permanent file based products or standard products in STEP 2 of this chapter. 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 user name INSTALL.

1. Put the new deadstart decks on the local file DSTDECK.

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

   If you did customize 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 this command:

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

Replace density with the tape density of the new deadstart tape. A tape request will be issued for an unlabeled tape with VSN=NDT. The tape may be assigned from the system console with the VSN,est,NDT command. 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.

OR

If you customized any deadstart binaries, create the new deadstart tape using this command:

```
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 tape with VSN=NDT. The tape may be assigned from the system console using the VSN,est,NDT command. 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:

- Created all new configuration files.
- Updated all existing configuration files.
- Customized any other released permanent files.

The files that normally reside on the special system user names NETOPS, SYSTEMX, LIBRARY, and SUBFAM0 (these are user names you cannot log in to from an interactive terminal) should be stored on user name INSTALL. Files that reside on login user names should have temporary file names. That way, the files will not be used until the system is deadstarted with the new release software.

In STEP 6 you will perform a two-phase installation: first you will use SYSGEN to automatically install files from the permanent file tapes; then you will move the files you have created and/or modified to their proper user names and file names. To eliminate unnecessary processing in phase 1, disable the loading of files you have modified.

To disable the loading of a file or files, make temporary modifications to the RECLAIM database so that it appears as if the file or files are not on the permanent file tape. That is, delete the file names from the database. Table 3-1 lists each product that is released with a permanent file, the user name where the file resides, and the corresponding PFG file name on the permanent file tape. Refer to table 3-1 to make a list of all the files you do not want loaded to the new system.

NOTE

If you have installed CDCNET, do not disable PFGCHA1. NAMSTRT has not yet been updated to include this product.

Once you have made a list of the files you do not want loaded to the new system, perform the following steps from a terminal under user name INSTALL:

1. Execute the RECLAIM utility by entering this command:

   ```
   RECLAIM.
   ```

   RECLAIM will respond 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 will prompt you for a list of file names:

   ```
   ENTER FILE NAMES.
   ?
   ```

3. Enter the desired file names:

   $\text{filename}_1,\text{filename}_2,\ldots,\text{filename}_n$

   Separate the file names with commas (,). If you cannot fit all the file names on one line, follow the last file name with a comma (,) and then 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 will respond with a report of the file names that have been deleted. Then, it will display the following prompt:

   ```
   ENTER DIRECTIVE.
   ?
   ```

4. Terminate the RECLAIM session by entering the following:

   ```
   END
   ```

   RECLAIM will respond with the following message:

   ```
   RECLAIM COMPLETE.
   ```

Here are some additional notes on using RECLAIM:

- You can list the names of all files that have been deleted 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:

  $\text{RECLAIM,Z./RESET,UN=NS2psrin,PF=*/filename}_1,\ldots,\text{filename}_n$

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 has been terminated.
- Ensure that all subsystem activity is terminated (except for the MAG and B10 subsystems).

If desired, perform a permanent file backup of your default family device using the PFDUMP utility. You can use the backup tapes if you need to restore any of the files. Refer to the NOS 2 Analysis Handbook for information about the PFDUMP utility.

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

To install new permanent files, follow the steps below. Enter the commands from the system console. 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.

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

   ```
   X.SYSGEN(FILES,vsn)
   ```

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

   SYSGEN will load files from the permanent file tapes and install them to their proper user names on the system. Once SYSGEN has completed installing files from the permanent file tape, you may proceed to install your site-modified files. Do this by executing the command in the next step as many times as necessary to install all of the modified files.

2. Move a file to its proper location from user name INSTALL by entering the following command from the system console:

   ```
   X.SYSGEN(MOVE,pfn,un,ct,ac,m)
   ```

Parameter	Description
pfn	Name of the permanent file to be moved. The file must reside on the installation user name INSTALL.
un	User name to receive files. Valid user names and passwords are listed in table 8-1.
ct	Specifies the permanent file category. Enter PU for public, PR for private, or S for semiprivate. The default is PR.
ac	Specifies the alternate user CATLIST option. Enter Y to allow alternate users to CATLIST the file or enter N to prevent alternate users from using CATLIST to list the file. The default is N.
m	Assigns read permission to user name INSTALL. Enter PERMIT to allow read permission. The default does not give read permission. PERMIT should only be used if ct=PR.

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

If you recompiled files NCFFILE and LCFFILE, follow the instructions for upgrading the files in the NAM 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 did not need to build a new deadstart tape to install your component order, your installation is complete; if you did write a new tape, continue to STEP 7: Deadstart the New System.

STEP 7: DEADSTART THE NEW SYSTEM

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.

Your component order installation is now complete.

CHAPTER 5

CORRECTIVE CODE INSTALLATION

CORRECTIVE CODE INSTALLATION 5

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 of this handbook. These instructions may be executed from an interactive terminal logged into the installation user name.

USING MODIFY FOR NOS AND NOS PRODUCTS

When corrective code has been 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 (which is either the name of the deck being modified or if more than one deck is being modified, the ident is NS2pmn where pmn is the previous modification number plus 1), 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 O.S.
*/      ****   NS23316.
*/      ****   REQUIRES - NONE.
*/      ****   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*
```

- Corrective Code. The corrective code to fix the problem consists of a DECK directive specifying the name of the deck to be modified followed by the assembler or compiler statements.

```
*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 MODESET.
```

Adding Corrective Code

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

- Apply the MODIFY modset to the NOS OPL by using the MODOPL procedure on file INSTALL and then run the appropriate build jobs.
- Place the corrective code on a USER file and run the appropriate build jobs.

The following two examples illustrate the two methods.

Creating a New Permanent OPL

This example illustrates how to install a sample modset named 1DS13. First, use the MODOPL procedure to apply the modset to OPLpsrin:

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

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

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

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

Putting Code on File USER

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 file USER is added only to the generated binaries.

For example, to build NOS with Indent 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 build procedure:

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

USING UPDATE FOR COMMON PRODUCT SET, NETWORK HOST PRODUCTS, CCP, AND CROSS

When corrective code has been 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 idents, 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: use UPDATE to create a NEWPL and then build the product; or 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 this code file to perform an UPDATE and then create a NEWPL:

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

If the update is successful, make the NEWPL permanent and run the build 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 File USER

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 of the Common Product Set and the Network Host Product build 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 build 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 CCPPH1, CCPBLB, and variant build jobs; for the CROSS build job, put the code on file UCRS. The build 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 build procedures. For more information, refer to the CCP/CROSS Permanent File section of the CCP section of chapter 7.

CDCNET BATCH CRITICAL UPDATE (BCU)

There are three potential types of corrections for CDCNET: a binary correction (BCU) to DCNS, a SOLVER correction to CHA1, and a problem that requires both a BCU and a SOLVER correction. 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 user name NETADMN and should be executed from an interactive terminal.

PRELIMINARIES

Before you begin the instructions, have the BCU tape available to be mounted and note the tape VSN and its density.

PROCEDURE

1. Log in to IAF under the user name NETADMN.

2. Install the BCU by entering the following procedure call:

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

Replace vsn with the VSN of the BCU tape located in the media number field of the external tape label; replace density with the tape density of the BCU tape (HY, HD, PE, or GE).

As this procedure is executing, your terminal displays a message indicating the CDCNET version level that is 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 complete normally. When the procedure completes, all new CDCNET software files are loaded to user name NETADMN. Configuration files, the exception list, INITDCN, and terminal user procedures are not loaded: these files are site-controlled files.

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 that you use a new ID=id parameter when you invoke the CHA1 build 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=INSTALL.
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 user name NETADMN:

```
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 of this manual.

6. Activate the new software and the message template file.

   a. Activate the new software by executing the SET_VERSION_LEVEL (SETVL) procedure in DCNPLIB. This procedure updates file INITDCN and the exception list to indicate the new default versions for all CDCNET software, namely, MANCC, ANACD, 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, consult the CDCNET section in chapter 7 of this handbook.

      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 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 2 Analysis Handbook).

      (2) Enter the following command:

          DROP,jsn.

          Replace jsn with the job sequence name of the Network Operator Utility (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 user name NETADMN. 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 user name NETADMN 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 correction tape is in RECLAIM format and is processed by the SYSGEN(UPGRADE) command. The correction tape includes:

- Corrected dual-state binaries for all NOS releases that support the NOS/VE system you are using. The Software Release Bulletin (SRB) for the BCU indicates which levels of NOS are corrected with the BCU.
- An updated NOS/VE dual-state library.

There are two options for installing a dual-state BCU:

- 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.
- Install the corrective code onto a version of NOS that was released prior to the version of NOS/VE you are using.

CORRECTING THE DUAL-STATE PARTNER

If you are running the NOS system that was released with the version of NOS/VE you are running, execute the following steps. However, before you begin:

- 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.)
- Have an unlabeled scratch tape available for mounting. 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 user name INSTALL.

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

   ```
   COMMON,SYSTEM.
   ```

2. Execute SYSGEN:

   SYSGEN,UPGRADE,SYSTEM,NEW,vsn,$density_1$,$density_2$.

   Replace vsn with the VSN of the BCU tape; replace $density_1$ with the tape density of the BCU tape; replace $density_2$ with the tape density of the new deadstart tape.

   The procedure will first request the BCU tape. After the BCU tape has been processed, the procedure will request the unlabeled scratch tape and write the new system to that tape.

CORRECTING AN EARLIER VERSION OF NOS

If you are running a NOS system that was released prior to the version NOS/VE that you are running, execute the following steps. Before you begin, have a 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 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.

   In order 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.

2. Enter this command:

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

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

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

3. Proceed to the Batch Critical Update section in the dual-state section of chapter 7.

CHAPTER 4

COMPONENT ORDER INSTALLATION

CHAPTER 6

CUSTOMIZING INSTALLATIONS

CUSTOMIZING INSTALLATIONS 6

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 following the steps in this chapter, note the following:

- Make a list of the products you want to customize. Next, check the product dependency chart (figure 6-1) to find out if other products are dependent on the products you want to customize and add those to the list. These are the products you must rebuild.
- Refer to chapter 7 for special information about each product you are installing.
- Special installation parameters, procedures, and commands relating to DECKOPL are detailed in chapter 9.
- Instructions for loading individual files (for example, the SRB, PSR report, or DECKOPL) are detailed 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.
- 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 have loaded the source installation tools, you can set up files for disk installation. Refer to chapter 9 for information about controlling the permanent file environment.

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, perform step 1; if you are customizing as part of an upgrade installation, perform step 2.

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

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

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

   NOTE

   If the source build procedures for the current level of NOS are 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 user name INSTALL:

   ```
   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 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 listing of DECKOPL by using the DECKLIS procedure. For information about additional DECKLIS parameters, refer to chapter 9.

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

   NOTE

   This listing is over 400 pages long.

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

   Whether installation will be from tape or disk.

   OR

   Where files reside for a disk installation.

   OR

   USER and CHARGE commands are used for executing installation procedures.

   OR

   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, and parameters that determine whether output PLs will be written.

   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 have finished, replace file COMMOD.

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

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

STEP 3: 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). Each file can contain the following:

- Programming Systems Report (PSR) corrective code from SOLVER. Refer to chapter 5 for detailed information about using 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 job, you can use the USERF parameter on the installation procedure call to specify the permanent file that contains your modifications. For example, you can store local modifications for the product AAM2 on the permanent file USERAAM. Then, when you are calling the AAM2 installation procedure to build AAM2, you can use the USERF parameter on the procedure call to pass the file name of the modification file USERAAM. For example:

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

NOTE

If you use file USER 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 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 1. If file SYSTEM is not local, REQUEST the released deadstart tape and COPY 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, 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,param2,....
```

Parameter	Description
DST=dstname	Local file name for your copy of the system. If the file dstname is not a mass storage file, dstname is copied to a mass storage file before the GTR commands in the SEED procedure extracts the required binaries. If DST=SYSTEM and no file named SYSTEM is local or on disk, SEED will perform a COMMON(SYSTEM) command for you.
REBUILD	The keyword parameter, REBUILD, causes SEED to load binaries into GLOBLIB and PRODUCT to satisfy all dependencies. You should specify REBUILD if you plan to customize only some products. If you will be building all products, you do not need to specify REBUILD.
RESET	The keyword parameter RESET causes SEED to initialize the files JOBSTAT and DAYFILS; these files keep statistics on the installation procedure jobs and dayfiles. To get a report of the resource usage of your installation jobs and a list of what jobs have passed or failed, refer to REPORT Procedure in 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 APL2 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 a 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 jobs. The operating system installation jobs use it to generate the compile file for operating system installations.

For execution of the MODOPL procedure, create file USER to include 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.

To execute the MODOPL procedure, enter this command:

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

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 build procedures, USER code affects only the binaries and not the new OPL.

STEP 6: BUILD PRODUCTS FROM SOURCE CODE

You build a product from source code by executing an installation procedure from file INSTALL. You can call the procedures interactively or you can submit them as batch jobs. If you call a procedure interactively, the procedure submits a batch job for execution (unless you have set IA=YES; refer to COMMOD file parameters in chapter 9). If you require file USER for an interactively called procedure, you must specify the USERF=pfn parameter (refer to STEP 3) on the call to the procedure.

Here is the format for calling an installation procedure:

BEGIN,procname,INSTALL,p_1,p_2,p_3,...,p_n.

Replace procname with the installation procedure name. These names are listed in tables 6-1 through 6-5 along with the unique parameters for the installation procedures. The optional parameters (p_1 through p_n) include two types of parameters: common and unique. Refer to COMMOD file parameters in chapter 9 for a list of common parameters; refer to chapter 7 for details about the unique parameters of a specific installation procedure.

Figure 6-1 shows the product dependencies and tables 6-1 through 6-5 list the OIP products and the DECKOPL build procedures that you need for assembling each product.

Product Dependencies

Some installation procedures require the output of another installation procedure. These procedures are dependent on other procedures; that is, they cannot be run until the procedure they depend on is completed. Figure 6-1 illustrates the product installation procedure dependencies.

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

NOTE

If a product has two possible installation procedures, the procedures for the product are enclosed in one box in the Build Dependency Chart (figure 6-1). Run only one procedure for each product.

Run Group 1

Run the group 1 procedures listed in table 6-1 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 6-2 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 6-1.)

Run Group 3

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

Run Group 4

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

Run Group 5

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

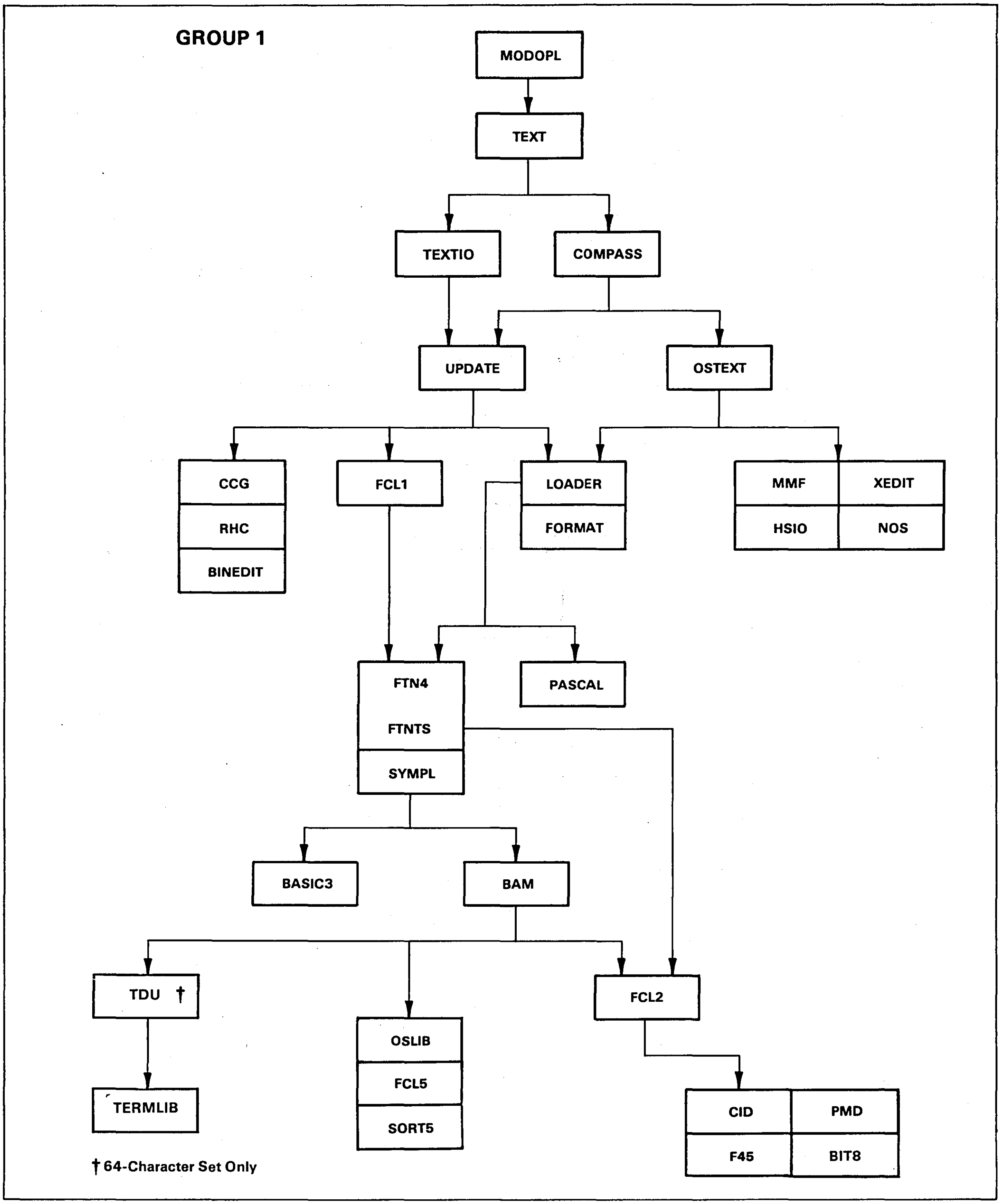

Figure 6-1. Build Dependencies Chart (Sheet 1 of 3)

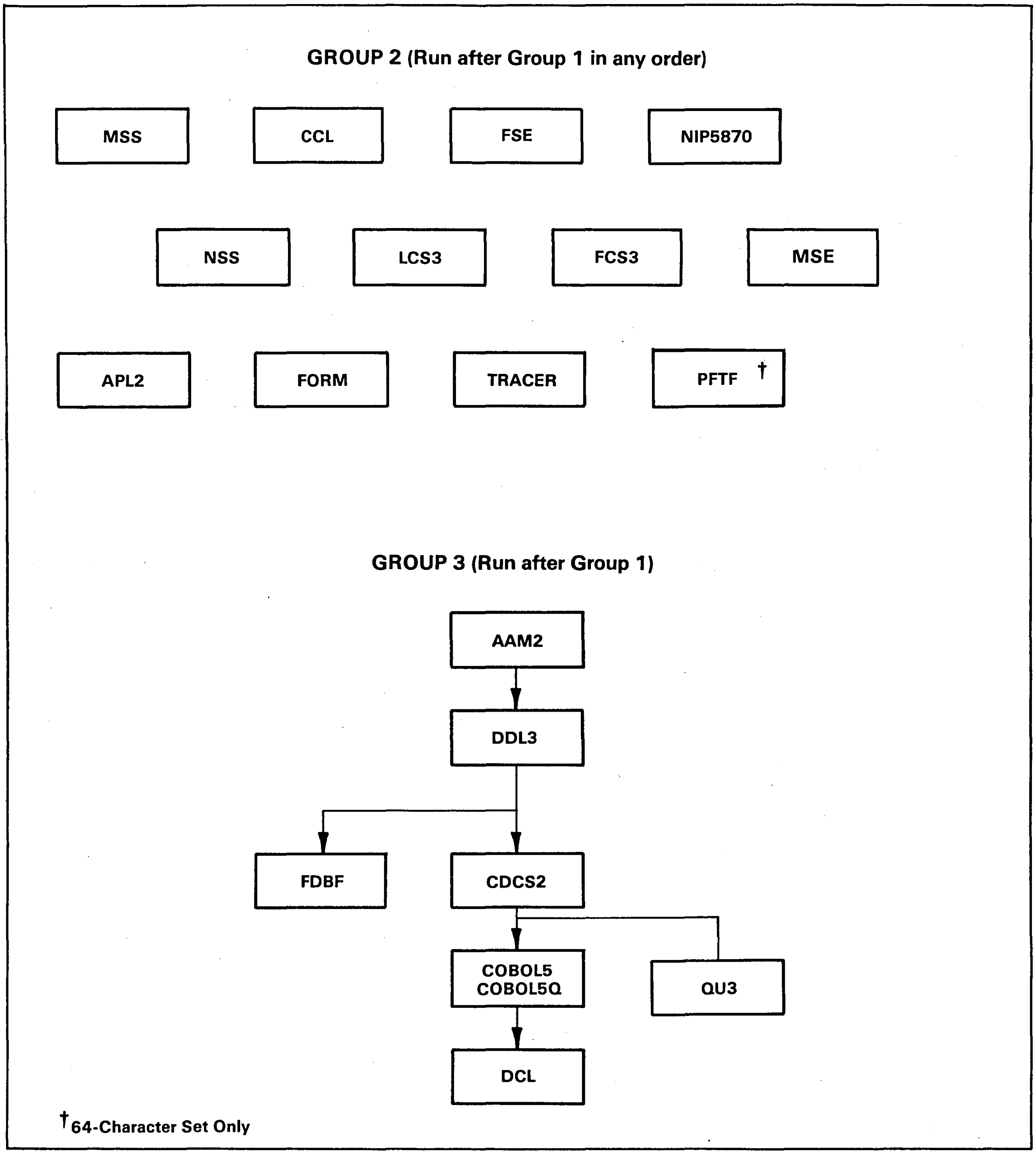

Figure 6-1. Build Dependencies Chart (Sheet 2 of 3)

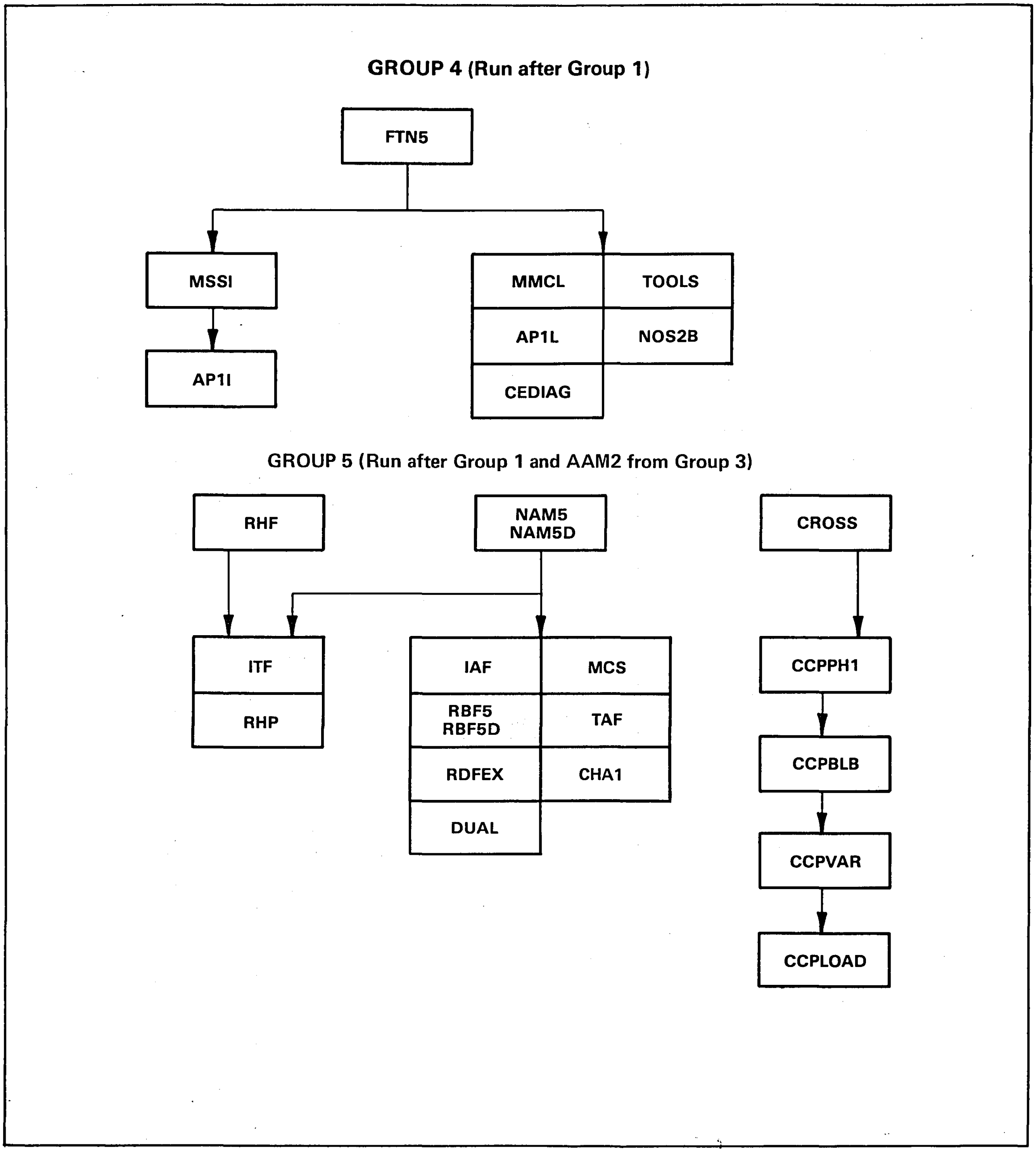

Figure 6-1. Build Dependencies Chart (Sheet 3 of 3)

Table 6-1. Group 1 Products (Sheet 1 of 2)

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

Table 6-1. Group 1 Products (Sheet 2 of 2)

Procedure Name	Unique Keywords	Required Files
FTNTS		FTI4psrin CPS1psrout
SYMPL		SYMPpsrin
PASCAL		PASCpsrin OPLpsrout
BASIC3		BAS3psrin CPS1psrout OPLpsrout
BAM		BAM1psrin TEXTpsrout
FCL5		FCL5psrin CPS1psrout
SORT5		SRT5psrin
OSLIB		OPLpsrout
FCL2		FCL4psrin CPS1psrout
BIT8		BIT8psrin
CID		CID1psrin OPLpsrout
PMD		PMD5psrin CPS1psrout
F45		F451psrin CPS1psrout
TDU†		TDU1psrin OPLpsrout
TERMLIB		OPLpsrout

† 64-character set only

Table 6-2. Group 2 Products

Procedure Name	Unique Keywords	Required Files
MSS	SAVELIB	OPLpsrout
CCL		CCL1psrin CPS1psrout OPLpsrout
FSE		OPLpsrout
NIP5870		OPLpsrout
NSS		NSS1psrin OPLpsrout
LCS3		LCS3psrin
FCS3		FCS3psrin
MSE	SAVELIB	OPLpsrout
APL2	TERMTYP=termtyp	APL2psrin OPLpsrout
FORM		FORMpsrin
TRACER		OPLpsrout
PFTF†		OPLpsrout TDU1psrin PFTFpsrin

† 64-character set only.

Table 6-3. 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 6-4. Group 4 Products

Procedure Name	Unique Keywords	Required Files
FTN5		FTN5psrin CPS1psrout CCG1psrout
MSSI	BLDMLIB MSAMLIB	MSSIpsrin OPLpsrout
AP1I	MEMSIZE	AP1Ipsrin
MMCL		MMCLpsrin
AP1L	MAPTYPE,ADD,MUL, DIV,SQRT	MMCLpsrin
CEDIAG		CEDGpsrin OPLpsrout
TOOLS		OPLpsrout
NOS2B		OPLpsrout

Table 6-5. Group 5 Products (Sheet 1 of 2)

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
CHA1	DEBUG,TRACE,ID=id	CHA1psrin NAM5psrout OPLpsrout
MCS	TRACE	MCS1psrin NAM5psrout OPLpsrout
TAF	DEBUG, TASKLB	OPLpsrout
CROSS		CRSSpsrin
CCPPH1	DIAG, REMT, BSTP, XTRAPLS	CCPBpsrin CCPTpsrin CCPRpsrin CCPDpsrin
CCPBLB	XREF	
CCPVAR	VN=vn	
CCPLOAD	GN=gn	

Table 6-5. Group 5 Products (Sheet 2 of 2)

Procedure Name	Unique Keywords	Required Files
	NOSLEV=noslev VELEV=velev	DUALpsrin OPLpsrout TDU1psrin BAM1psrout

Table 6-6. OIP Options and Associated Build Procedures (Sheet 1 of 3)

Product Name	Related Build Procedures
APL 2	APL2
BASIC 3	BASIC3
COBOL 5	COBOL5/COBOL5Q, FCL1, FCL2, PMD
Communications Control Program (CCP) 3	CCPPH1, CCPBLB, CCPVAR, CCPLOAD
Control Data Distributed Communication Network (CDCNET)	CHA1
Concurrent Maintenance Library (CML)	No build procedure. Consult the Concurrent Maintenance Library Reference Manual.
Conversion Aids Subsystem 3	LCS3, FCS3
CYBER Cross System 1	CROSS
CYBER Database Control System 2	CDCS2
CYBER Interactive Debug	CID
Data Catalogue 2 Version 2.0	DCL
Data Description Language	DDL3
FORTRAN Database Facility 1	FDBF
FORTRAN Extended 4	FTN4, FCL1, FCL2, PMD
FORTRAN Extended 4 with Interactive Option	FTNTS, FCL1, FCL2, PMD
FORTRAN 5	FTN5, FCL5, PMD, CCG
FTN 4/5 Conversion Aid 1	F45

Table 6-6. OIP Options and Associated Build Procedures (Sheet 2 of 3)

Product Name	Related Build Procedures
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, CEDIAG, FORMAT, SYMPL
Mass Storage Extended	MSE
Mass Storage Subsystem 1	MSS
MAP Macro Control Library	MMCL
Message Control System 1	MCS
MMCL V3,	
MSSI V3, MAP III	MSSI, AP1I, AP1L
Multi-Mainframe Module 1	MMF
Network Access Method 1	NAM5/NAM5D
NOS Online Manuals	No Build Procedure
NOS Scope 2 Station	NSS
NOS 2 Package	LOADER, COMPASS, BAM, BIT8, OSTEXT, UPDATE, FORM, AAM2, TEXT, TEXTIO, BINEDIT, CCL, RDFEX, OSLIB, NOS, NOS2B, NIP5870, TDU, TERMLIB, PFTF
PASCAL 170	PASCAL
Printer Support Utility	PSULINK
PTF/QTF File Transfer Facilities	RHP, RHC
Query/Update 3	QU3
Remote Batch Facility 1	RBF5/RBF5D
Remote Host Facility Access Method	RHF, RHC
Sort/Merge 5	SORT5
Tracer 1	TRACER

Table 6-6. OIP Options and Associated Build Procedures (Sheet 3 of 3)

Product Name	Related Build Procedures
Transaction Facility 1	TAF
Xedit 3	XEDIT
3270 TIP for CCP	(Part of CCP 3)

STEP 7: CREATE A NEW DEADSTART TAPE

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

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

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

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

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 into SYSPROC, which is always called after deadstart. Refer to the NOS 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.

You can modify the permanent files 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 the file from the tape and break it into its individual component pieces. For example, SYSGEN(MSE1) will load file PFGMSE1 from the tape and create files MSESLAV, MSE, and MSECONF.

Table 3-1 (which shows files for which you can disable loading) and appendix B (which lists 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. For an upgrade or component order installation, return to chapter 3 or 4 to finish the installation. For a first-time installation, refer to SYSGEN(MOVE) in chapter 8 for information about moving permanent files.

CHAPTER 7

SPECIAL PRODUCT INSTALLATION INFORMATION

SPECIAL PRODUCT INSTALLATION INFORMATION 7

This chapter gives information about subsystem initiation and 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. The following installation procedures are discussed in this section:

SUBSYSTEM INITIATION

SUBSYSTEM STARTUP FILES

When you install a subsystem, you must ensure that there is a startup procedure file for initiating the subsystem. (Table 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:

subxxxx

sub The 3-character name for the subsystem (for example, CDC).

xxxx An optional 0- to 4-character extension to the name (for example, CDCS2).

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

SUBSYSTEM ENABLE AND DISABLE COMMANDS

In addition to the procedure file creation and placement, you must issue ENABLE commands to tell the system 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 necessary ENABLE and DISABLE commands.

AUTOMATIC SUBSYSTEM INITIATION

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

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

- Enable the subsystem and any other system functions by adding the appropriate ENABLE entries to the IPRDECK. (The released IPRDECKs contain default ENABLE commands for all the ordered products, 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 (Sheet 1 of 2)

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

† For complete information on automatic initiation of dual state, refer to the Dual State section of this chapter.

Table 7-1. Subsystem Initiation (Sheet 2 of 2)

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

NOTES:

- RDF and MAG are always added to the deadstart file for system maintenance purposes. All other procedures are placed only on permanent files.
- MSE, MSS, and RDF may require additional IPRDECK entries (refer to the NOS 2 Analysis Handbook).
- For more information about the IPRDECK, ENABLE and DISABLE commands, and subsystem startup, refer to the NOS 2 Analysis Handbook.
- Enable either IAF or RDF, not both. To help maintain access security to RDF, enable RDF only when needed as specified in the NOS On-Line 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 and add *PROC directives to your LIBDECK. For more information, refer to the NOS 2 Analysis Handbook.
- n stands for control point number.

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 file USER. (Without this directive, the system gathers only normal file statistics.)

ADDITIONAL PROCEDURES

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

Each routine must have an entry point of the form CMPR$nn (nn is two decimal digits within the range 11 through 63). The 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	Entry point name, 6-bit display code, left-justified, zero fill.
	17 through 0	1.
1	59 through 18	0.
	17 through 0	Starting address of compression code.
2	59 through 18	0.
	17 through 0	Starting address of decompression code.

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

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

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

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

```
GENTBL name
```

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

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

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

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

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

APL2 - APL VERSION 2

The installation of APL2 consists of building APL2 and then installing APL2 permanent files. This 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 the BEGIN,APL2,INSTALL... command. The file must contain only a one-line message which cannot exceed 80 characters.

SYSGEN FUNCTIONS

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

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

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

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

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

   ```
   X.DIS.
   USER(SYSTEMX,SYSTEMX)
   ATTACH(PFGAPL2/UN=INSTALL)
   SYSGEN(APL2)
   ```

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

APL VALIDATION REQUIREMENTS

User names APL0 and APL1 are required for running APL. If you have no validation file, SYSGEN will automatically create the user names and assign default passwords. If you have a validation file, you must create user names APL0 and APL1 with the limits shown in table 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 Administration Handbook for information about using MODVAL and about changing validation files. If you want to use SYSGEN to reload files on these user names, you must temporarily set the passwords to the default values or you must update SYSGEN.

Table 7-3. Recommended Limits for APL0 and APL1

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

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

BAM - CYBER RECORD MANAGER BASIC ACCESS METHODS VERSION 1

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

Parameter	Default	Significance
#LBLIM#	10D	Number of words in tape label buffer. Because each user label requires 9 words, set LBLIM to 9m+1; m is the maximum number of file header (HDRn) labels allowed. Minimum value is 10 words.
#CMU# #NOCMU#	1 1	Specifies use of compare and move unit (CMU) instructions in routine MOVE$RM. To remove the CMU code, delete the definition of CMU. To remove the no CMU code, delete the definition of NOCMU. If CMU and NOCMU are both defined, CYBER Record Manager determines at run time which MOVE routine to use by checking the CMU flag in RA.CMU. The use of CMU instructions reduces the execution time of a program using CYBER Record Manager for records of over 40 characters. The use of CMU instructions in programs to be executed on models 835, 840, 845, 850, 855, 860, or 990 is not recommended.

BASIC3 - BASIC VERSION 3

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

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

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. Because installation procedures use the default values, changing the values is not recommended.

Parameter	Default	Description
IP.FPC	10	Maximum number of characters in a keyword for a procedure call or procedure header directive. Maximum value is 10.
IP.SCS	40	Maximum number of characters for default and actual values. Maximum value is 80.
IP.LCS	10	Maximum number of characters in a label string. Maximum value is 10.
IP.PNL	50	Procedure nesting limit. Maximum value is 1023.
IP.FP	50	Maximum number of keywords in a procedure call or procedure header directive. Maximum value is 500.
IP.DPF	1	Flag indicating logical existence of default procedure file name. Value / Definition 0 — No default procedure file name. 1 — Procedure file name defaults to value of IP.DPFN.
IP.DPFN	PROCFIL	Default procedure file name.
IP.TAPO	1	Flag indicating whether a procedure can reside on tape. Value / Definition 0 — Procedure file cannot reside on tape. BEGIN hangs in RECALL if execution from tape is attempted. A value of 0 decreases the execution size of CCL by 700_8 words for BEGIN, REVERT, WHILE, and ENDW. 1 — Procedure file can reside on tape.
IP.EXP	100	Number of operands and operators allowed in a CCL expression. For each unit that this parameter is decreased from 100, the execution size of CCL is reduced by 2 words.

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. In order to attach the requested procedure file, the system searches the indirect access files first and then the direct access files.
IP.NPV	6	Value used in the calculation of the size of the pattern value table (PVT). The PVT stores the checklist entries for each parameter in the procedure headers. The following formula determines the size of the PVT in words. PVT = IP.NPV * IP.FP * 2
IP.RLD	1	Flag indicating whether the system does a sequential or random search of a library to find the requested procedure. A random search is usually faster than a sequential search. Value — Definition 0 — Search library sequentially. 1 — Search library randomly by using the library directory.
IP.SCL	150	Maximum length in characters of lines in a procedure. Any restrictions as to the length of a command remain in effect, but a comment following the command terminator may extend to the length specified by IP.SCL.

ADDITIONAL PROCEDURES

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

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

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

CCP - CYBER CROSS SYSTEM AND COMMUNICATIONS CONTROL PROGRAM

This section describes the following:

- Hardware Requirements
- Configuration Information
- Released Software Variants
- Build Steps Description
- Binary Installation Option
- General Build Step Call
- 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/Cross Permanent Files
- 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_8 is required to build CCP. The following equipment configuration is the minimum required to execute CCP:

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

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

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

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 for each NPU is referenced by the VARIANT parameter on the NPU statement in the NDL file. The NLFFILE is stored as a direct access permanent file on user name NETADMN. It must be a public, semiprivate, or private files with read permission given to user name NETOPS.

An NLFFILE which contains a number of different 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 must customize your own variants and load file, the load file generated by the CCPLOAD procedure must be moved from user name INSTALL to NETADMN. Before the file is moved, it should be renamed from Gxxx (where xxx 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 select, you will 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 describes the released CCP variants.

Table 7-4. Released CCP Variants

Variant Name	NPU Size	HIP	LIP	Buffer Space	TIPS
V1F	96K	YES	NO	42K	ASYNC, HASP, BISYNC
V2F	96K	YES	NO	42K	ASYNC, MODE4
V3F	96K	YES	NO	42K	ASYNC, X.25 A-A, X.25 PAD
V4F	128K	YES	NO	37K	ASYNC, HASP, MODE4, BISYNC X.25 A-A, X.25 PAD
V1L	96K	YES	YES	37K	ASYNC, HASP, BISYNC
V2L	96K	YES	YES	42K	ASYNC, MODE4
V3L	96K	YES	YES	40K	ASYNC, X.25 A-A, X.25 PAD
V4L	128K	YES	YES	42K	ASYNC, HASP, MODE4, BISYNC X.25 A-A, X.25 PAD
SM1	65K	YES	NO	24K	ASYNC
SM2	65K	YES	NO	19K	ASYNC, HASP
SM3	65K	YES	NO	15K	ASYNC, BISYNC
SM4	65K	YES	NO	18K	ASYNC, MODE4

Table 7-5 summarizes each of 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 7-5. Options and Variants with CCP

Option	Variants Received
A	V1F, V2F, V3F, 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

When SYSGEN is executed, a compiled NDL (NCFFILE and LCFFILE) will be placed on user name NETADMN. The source for the NDL is placed on user name NETADMN as file NDLDATA. 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). The NDL for options A and B specifies SM1.

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 file.
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.

†Terminal Interface Program.

Build Step	Description
CCPBLB	Updates the PCMB program library with temporary user-supplied corrective code from file UCCP and generates the CCP object code library (BCMB); writes an updated version of CCPB. This build step is also called the CCP full compile and assembly build step.
CCPVAR	Updates the PCMB program library with temporary user-supplied corrective code from file UCCP and generates a CCP variant load module (Zvvv) and program initiation control block (PICB) file (Ivvv) from the BCMB file according to the user-specified variant definitions in file USERBPS; writes Vvvv. This build step should be repeated for each 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) via program LFG (refer to the NOS 2 Analysis Handbook). The load file includes the phase 1 and dump load modules (file ZMUX) and system autostart load module (file ZSAM) from step CCPPH1, and the variant load modules (Zvvv) and PICBs (Ivvv) from step CCPVAR.
CCPPURG	Purges the noncritical permanent files created by the other build steps. It does not purge the load file from build step CCPLOAD and the user-supplied files. This build step is not required; it is only a cleanup utility. However, since previous build steps do not purge the noncritical permanent files, it is suggested that CCPPURG be run to make more disk space available.

The final result of the CCP/Cross build steps is the generation of a NAM network load file (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 build process. Figure 7-3 illustrates the relationship of build steps to critical files and tapes involved in CCP installation.

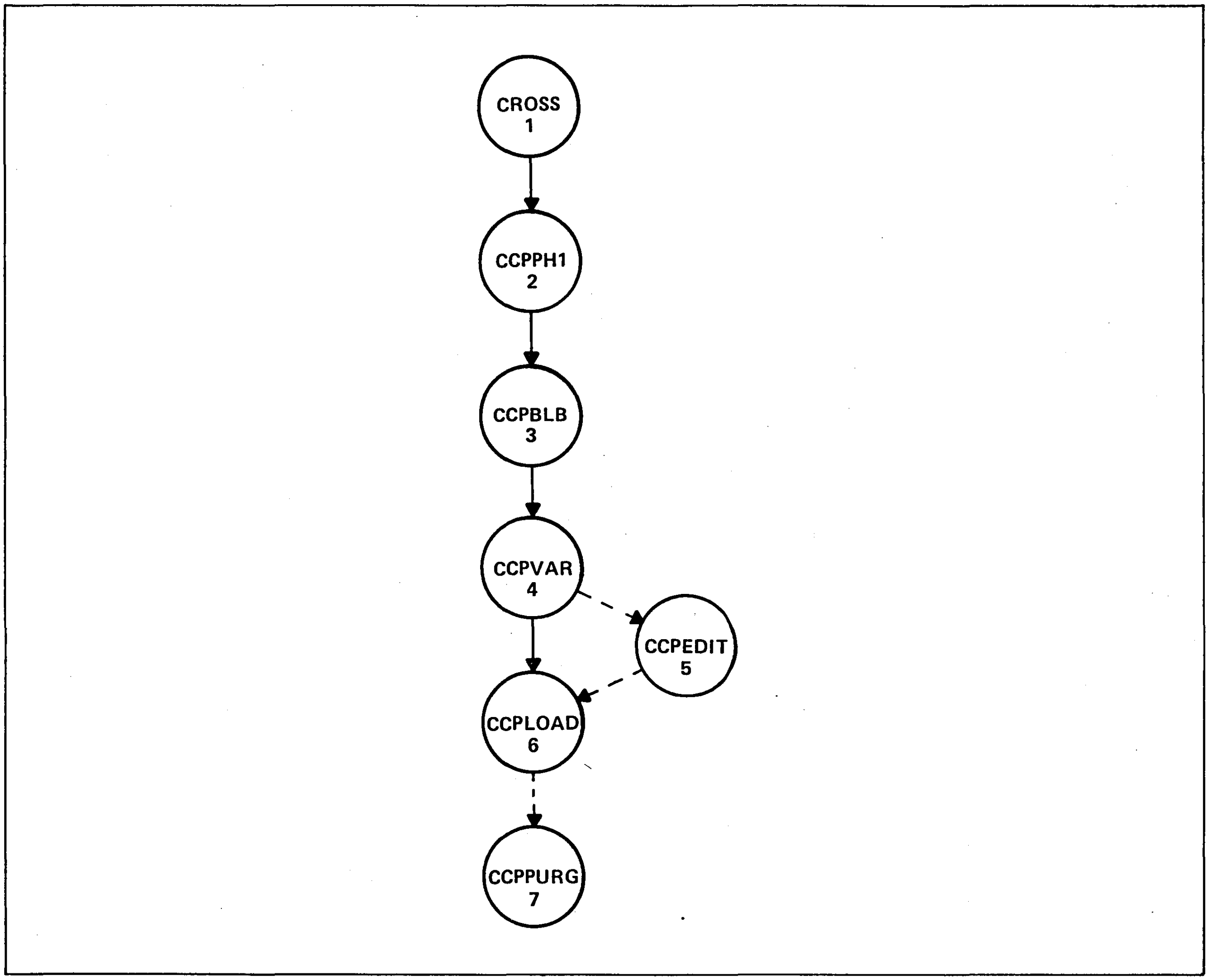

Figure 7-1. CCP/CROSS Build Step Dependencies

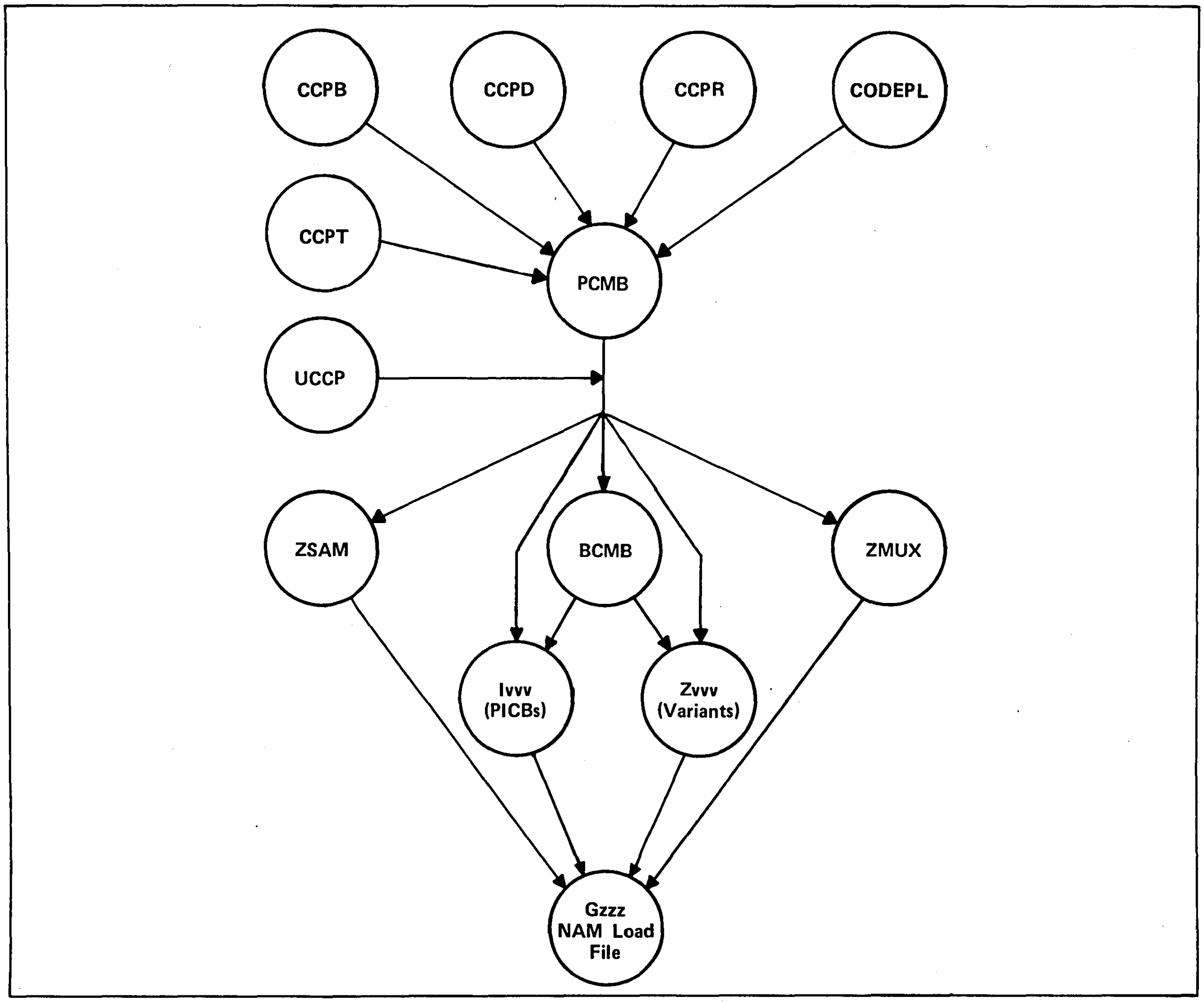

Figure 7-2. CCP File Dependencies

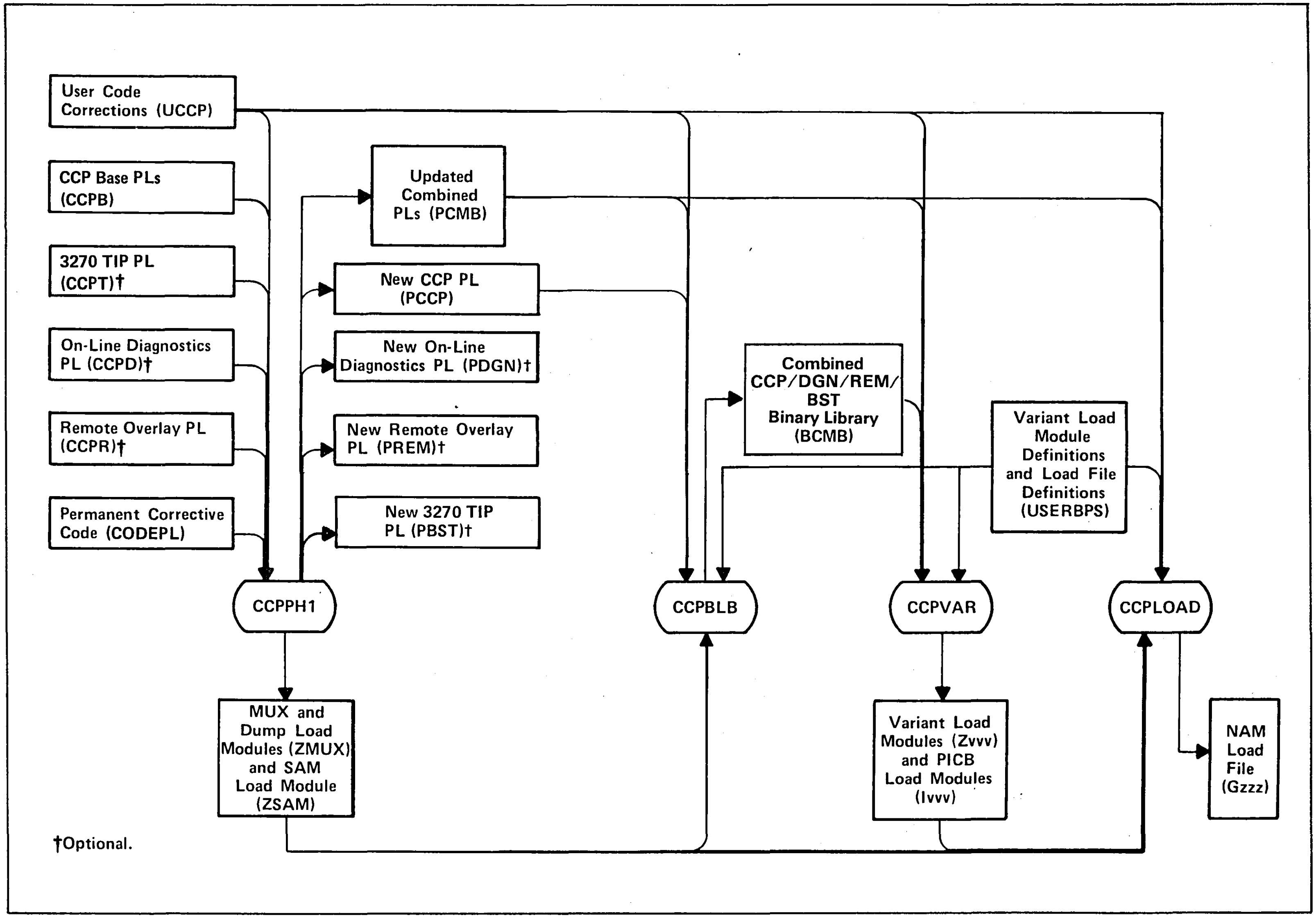

Figure 7-3. Integration of Program Libraries in CCP Build Process

BINARY INSTALLATION OPTION

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

- All permanent files necessary to begin the build process with the CCPVAR step.
- Several CCP variants and a load file. (The 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 the omitted build steps and you do not need to add online diagnostics or the 3270 TIP. (If you ordered the Remote Concentrator Products, they are included in the CCP files.)

To perform a CCP binary installation, enter:

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

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

CCP/Cross uses the same build process as other products for declaring auxiliary pack assignments, USERCHG information, and job origin type. Refer to the COMMOD File Parameters section in chapter 9.

The format of the BEGIN command follows:

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

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

Parameter	Description
OLD=xxx	Specifies CCP variant to be patched; used only with CCPEDIT. Supply the 3-character alphanumeric name.
REMT=remt	Specifies whether remote concentrator products are present (CCPR); used only with CCPPH1. If this parameter is omitted, REMT=NO is assumed.
VN=vvv	Specifies a variant name that matches the variant name in the VRD definition in USERBPS; used only with CCPVAR.
Vx=vvv	Specifies the variant name that was used in CCPVAR; used only with CCPPURG. x is an integer within the range 1 through 10.
XREF=xref	Specifies whether the build step generates a cross-reference listing of the Pascal source of CCP; used only with CCPBLB. If this parameter is omitted, XREF=NO is assumed.
XTRAPLS=xtrapls	Specifies whether extra program libraries are to be merged into the PCMB; used only with CCPPH1. If this parameter is omitted, XTRAPLS=NO is assumed.

SECURITY CHARACTER PARAMETERS

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

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

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

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

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

CROSS - CROSS SYSTEM INSTALLATION

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

```
BEGIN,CROSS,INSTALL,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, later in this section). For a description of the Cross installation parameters that can be changed, refer to installation parameters for CROSS, later in this section.

The CROSS build step creates the following output files.

File	Description
AASM	Cross macro assembler.
ACYP	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 file.
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 77000_8 central memory (CM) words. Some programs require a Pascal compiler that accommodates more than 1536 global symbol definitions; for example, CCP requires 6144 global symbols. Increasing the size of the global symbol table without increasing the in-core symbol table, however, results in a significant increase in compilation time. Further, an increase in the number of CM words must accompany any increase in the size of the in-core symbol table (4 CM words per symbol table entry).

CCPPH1 - CCP PHASE 1

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

```
BEGIN,CCPPH1,INSTALL,DIAG=diag,REMT=remt,BSTP=bstp,XTRAPLS=xtrapls,CCPLIST=option,
NOECS,NOPURGE.
```

The CCPPH1 build step uses the following input files.

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

CCPPH1 generates the following output files.

File	Description
AALK	Autolink program binary.
AEXP	Build parameters expand program binary.
CCPDpsrout	New diagnostic file (if DIAG=YES).
CCPRpsrout	New remote file (if REMT=YES).
CCPTpsrout	New 3270 TIP file (if BSTP=YES).
LIMC	Listing of Expand and Autolink programs (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.

File	Description
PDGN	New diagnostic program library (if DIAG=yes).
PREM	New remote program library (if REMT=yes).
SMUX	Dump bootstrap symbol table.
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 library.
AASM	Cross macro assembler.
ACYP	CYBER Computer Systems Pascal compiler (if XREF=YES).
AEXP	Build parameters expand program binary.
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	Dump bootstrap symbol table.
UCCP	Optional user corrective code. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
USERBPS	User variant build parameters file. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.

CCPBLB produces these output files.

File	Description
BCMB	Combined CCP/diagnostics/remote/3270 TIP binary library.
CCPBpsrout	CCP release file.
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 (phase 2) 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 parameters.

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

CCPVAR requires the following input files.

File	Description
AALK	Autolink program library.
AEDT	MPEDIT program.
AEXP	Build parameters expand program library.
AFMT	Pascal binary output formatter program.
ALNK	MPLINK program.
APAS	MP17 Pascal compiler.
BCMB	Combined CCP/diagnostics/remote/3270 TIP binary library.
PCMB	Updated combined program library.
SMUX	Dump bootstrap symbol table from CCPPH1 build step.
UCCP	Optional user corrective code. Refer to User-Supplied Files, in CCP/Cross Permanent Files, later in this section.
USERBPS	User variant build parameters file. Refer to User-Supplied Files, later in CCP/Cross Permanent Files, later in this section.

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

File	Description
Ivvv	PICB load module.
Lvvv	Variant load module and PICB listing (if requested).
Svvv	Symbol table for CCP variant load module Zvvv and PICB load module Ivvv.
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 CYBER Cross System Build Utilities Reference Manual). The CCP build process requires this step only for those cases where there is a minor difference between an existing load module and the desired load module. Refer to General Build Step Call, earlier in this section, for descriptions of the parameters.

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

CCPEDIT requires four input files.

File	Description
AEDT	MPEDIT program.
$Svvv_1$	Symbol table associated with variant load module vvv_1.
UEDZ	Optional direct or indirect access permanent file of MPEDIT directives to patch a CCP variant load module. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
$Zvvv_1$	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 CCPLOAD 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 library.
Ivvv	Previously created PICB load module(s).
PCMB	Updated combined program library. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
UCCP	Optional user corrective code. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
USERBPS	CCP load file definitions file supplied by user. Refer to User-Supplied Files in CCP/Cross Permanent Files, later in this section.
ZMUX	MUX firmware and dump bootstrap load module.
ZSAM	System autostart load module.
Zvvv	Previously created CCP variant load module(s).

CCPLOAD generates one output file.

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

Relocate file:

If the released version of the NS job skeleton JOBNS is used (refer to NAM5 - Network Access Method Version 1 in this chapter), rename Gzzz file as NLFFILE and move NLFFILE to user name NETADMN. NLFFILE should be a direct access file with read permission given to user name NETOPS.

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 (G_{zzz}). Refer to General Build Step Call, earlier in this section, for descriptions of parameters.

BEGIN,CCPPURG,INSTALL,V1=vvv_1,V2=vvv_2,...,Vn=vvv_n.

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

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/3270 binary library.

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

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

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

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

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

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

NOTE

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

User-Supplied Files	Description
UCCP	Optional direct or indirect access permanent file of user code corrections to CCP. The contents of this file should be the same for all build steps requiring it.
UCRS	Optional direct or indirect access permanent file of user code corrections to Cross. This file 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 the listing of the permanent files.

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

NOTE

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

USERBPS File

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

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

$\text{keyword}_1\text{=value}_1\text{,keyword}_2\text{=value}_2\text{,...,keyword}_n\text{=value}_n\text{.}$

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

$\text{keyword}_1\text{=value}_1\text{/value}_2\text{/.../value}_n\text{.}$

This is equivalent to the following:

$\text{keyword}_1\text{=value}_1\text{,keyword}_1\text{=value}_2\text{,...,keyword}_1\text{=value}_n\text{.}$

The following syntax rules apply to all definitions.

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

CCP SYSTEM DEFINITION

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

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

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

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

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

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

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

Keyword	Description

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

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

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

CCP VARIANT LOAD MODULE DEFINITION

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

The format follows:

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

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

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

Keyword	Description
	The following values are optional.

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

Examples:

VT=L/D/T

VT=F/D

VT=R

Keyword	Description
SZ=xK	Specifies variant memory size: 65K, 81K, 96K, or 128K (x is a 2- or 3-digit number).
TS=t1/t2/...	Specifies which Terminal Interface Programs (TIPs) are to be included in this variant. TS can assume up to 10 different order-independent values.

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

Example 1:

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

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

Example 2:

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

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

Example 3:

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

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

CCP LOAD FILE DEFINITION

The format follows:

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

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

Example 1:

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

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

Example 2:

LFD=EX5,LM=EX3.

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

CCP EXTRA PLs DEFINITION

The format follows:

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

Keyword	Description
PLS	Identifies entry as an extra PL definition and specifies which user-supplied PLs should be merged into PCMB. 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 and two NPUs.

- All examples require 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 assignment(s) and/or NDL name for major components in the network as chosen for this example. In the configuration shown in figure 7-4:

- 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 build job. At the end of the job, the files are copied to the new release file and purged. XREF=YES generates a Pascal cross-reference listing.

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

   a
   BEGIN,CCPVAR,INSTALL,VARLIST=PF,VN=<u>VNA</u>.

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

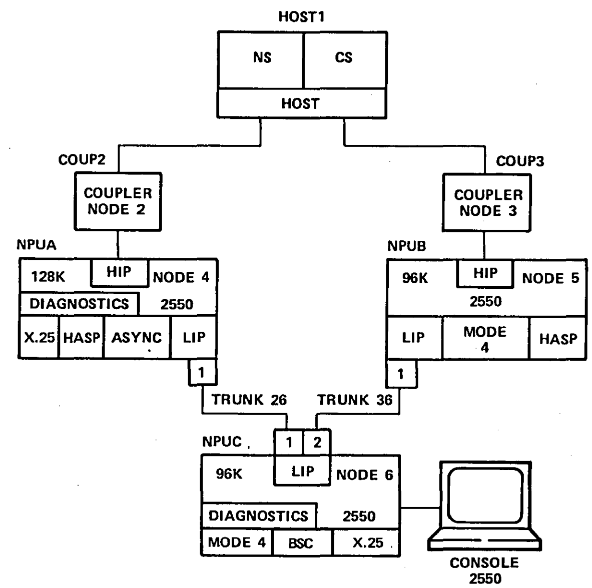

Figure 7-4. Network Configuration - Example 1

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

    BEGIN,CCPVAR,INSTALL,VN=VNB. (b)

    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. (c)

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

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

    BEGIN,CCPLOAD,INSTALL,GN=EX3. (n)

    The load file name is GEX3.

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

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
     SUPLINK  LLNAME=LL35.                             INPUT

     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 (Sheet 1 of 2)

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

     SUPLINK  LLNAME=LL26.

     SUPLINK  LLNAME=LL36.

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

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

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

Figure 7-5. USERBPS Definitions, NDL Source Input, and EQPDECK Entries for Example 1 (Sheet 2 of 2)

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.
```

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 and an X.25 line.
- 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, EQPDECK entries, and NDL source input.

2. Follow steps 2 through 6 in example 2.

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

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

   The load file name is GEX4.

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

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

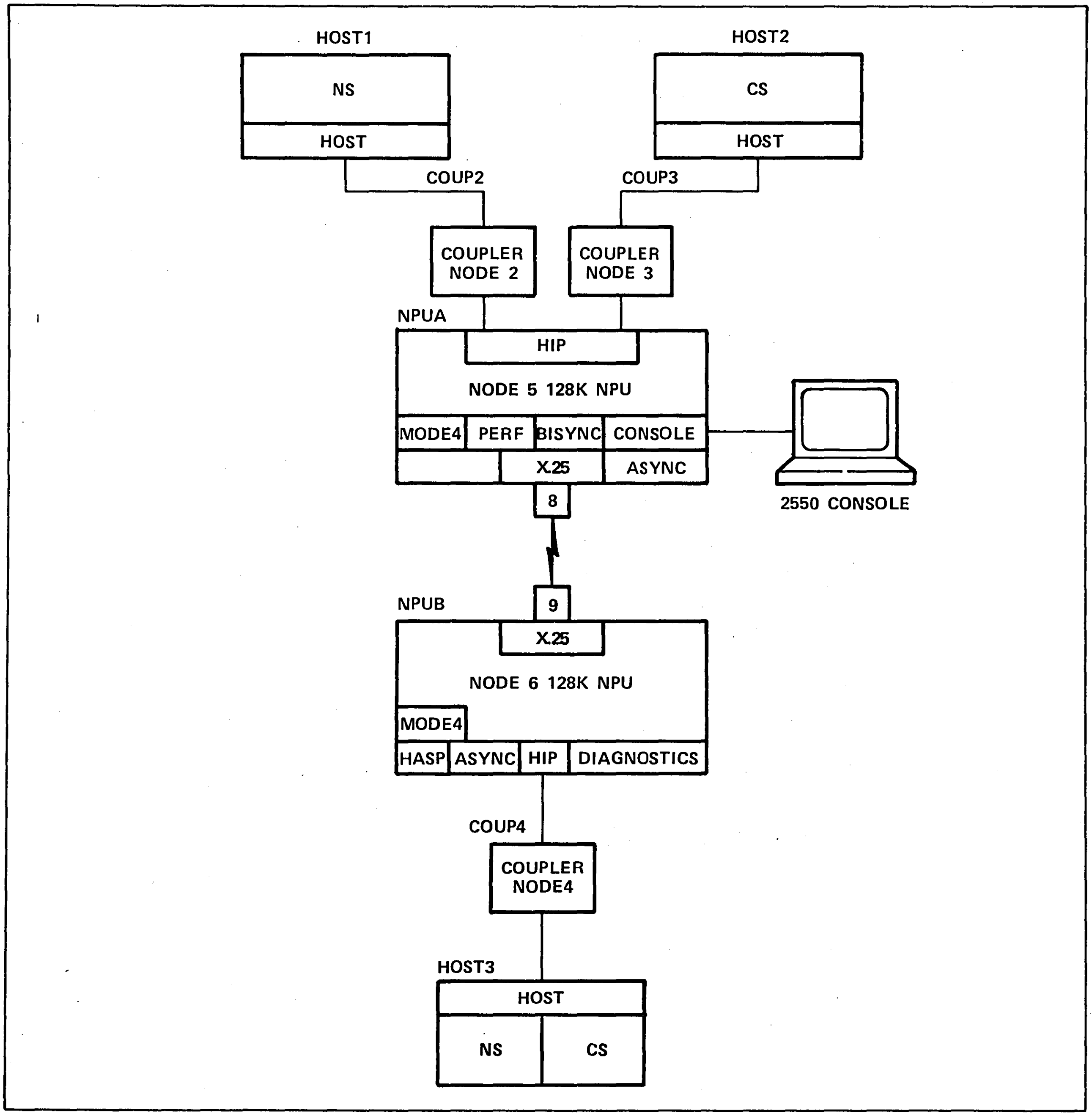

Figure 7-6. Network Configuration - Example 2

```
*   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                    USERBPS
*       XP     X25 PAD                       Definitions
*
*   c d e f    g h i j k  m
SYS=C/D/P/R,TS=A/B/H/M/XA/XP.
*
*     VARIANT DEFINITIONS
*
*         NPU A
*
*     a     c   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.
*
*
```

Figure 7-7. USERBPS Definitions, NDL Source Input, LCF Source Input, and EQPDECK Entries for Example 2 (Sheet 1 of 5)

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



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

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

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

Figure 7-7. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 2 (Sheet 2 of 5)

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



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


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


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

INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,DBL=7,ABL=7,DBZ=1024.
OUTCALL,NAME1=QTFS,PID=M02,SNODE=2,DNODE=3,DBL=7,ABL=7,DBZ=1024.
INCALL,FAM=0,UNAME=NETOPS,ANAME=PTFS,DBL=7,ABL=7,DBZ=1024.
OUTCALL,NAME1=PTFS,PID=M02,SNODE=2,DNODE=3,DBL=7,ABL=7,DBZ=1024.
INCALL,FAM=0,UNAME=NETOPS,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=M03,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=NETOPS,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=M03,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, LCF Source Input and EQPDECK Entries for Example 2 (Sheet 3 of 5)

```
LCFFILE:LFILE.

TITLE   LCFFILE FOR HOST 2


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


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

**
*
*    INCALL/OUTCALL BLOCKS FOR HOST 2

*
**



INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,DBL=7,ABL=7,DBZ=1024.
OUTCALL,NAME1=QTFS,PID=MO1,SNODE=3,DNODE=2,DBL=7,ABL=7,DBZ=1024.
INCALL,FAM=0,UNAME=NETOPS,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=NETOPS,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=NETOPS,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, LCF Source Input and EQPDECK Entries for Example 2 (Sheet 4 of 5)

```
LCFFILE:LFILE

TITLE   LCFFILE FOR HOST 3


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


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


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


INCALL,FAM=0,UNAME=NETOPS,ANAME=QTFS,PORT=9,SNODE=6,DNODE=4,DBZ=1024,DBL=7,ABL=7,
   UBZ=1024,UBL=7,WS=7,DPLS=1024.
OUTCALL,NAME1=QTFS,PID=M01,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=NETOPS,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=M01,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=M02,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=M02,SNODE=4,DNODE=6,SHOST=2D3032,PORT=9,DHOST=3,DBZ=1024,DBL=7,ABL=7,
   UBZ=1024,UBL=7,WS=7,DPLS=1024.
```

Figure 7-7. USERBPS Definitions, NDL Source Input, LCF Source Input and EQPDECK Entries for Example 2 (Sheet 5 of 5)

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

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

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

CDC PROCEDURE FILE

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

SPECIAL NOTES

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

ACCOUNTING TABLE

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

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

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

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

Field 1 (column 1)   Blank or comma.

Field 2 (column 2)   One of the following user request codes.

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

Field 3 (column 11)   The macro identifier ACC.

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

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

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

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

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

CDCNET - CONTROL DATA DISTRIBUTED COMMUNICATIONS NETWORK

This section contains information for installing and configuring CDCNET. CDCNET is composed of two subproducts: DCNS and CHA1. DCNS is a binary-supported product that consists of the DI ANACD, MANCC, and NPA software. CHA1 consists of the NAM applications and utilities which interface with the DCNS software.

General CDCNET information is documented in the following subsections:

- Configuration Information
- Installing CDCNET for the First Time
- Upgrading CDCNET
- Activating a New Network

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

- USER File Directives
- Message Templates
- CDCNET Host Application (CHA1) Product Content
- Unique Parameters for CHA1
- CDCNET Network Host Application Requirements

CONFIGURATION INFORMATION

- CDCNET Configuration Files. There are several types of configuration files used by CDCNET:

  - Device Interface (DI) system configuration files define the hardware and software attributes of the different types of DIs.
  - Terminal User Procedures (TUPs) and Terminal Definition Procedures (TDPs) are used to specify additional information about lines and terminal devices in addition to the information specified in the system configuration files.
  - 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 files, TDPs, TUPs, and the exception list file have both a CDCNET file name and a NOS permanent file name. The network Directory File (NETDIR) associates the CDCNET name with the NOS name. Refer to the CDCNET Configuration and Site Administration 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 user name NETADMN except for the network directory file (NETDIR) which resides on user name NETOPS. You can log in to user name NETADMN to perform the configuration of CDCNET (the released password is NETADMN). Both the NETADMN and INSTALL user names have been given write permission to the NETDIR file on NETOPS. This permission allows you to update this file from an interactive terminal.

For more information on configuring CDCNET, refer to the CDCNET Configuration and Site Administration 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 in this section. For information on compiling a new NDL, refer to the NAM section of this chapter.

- ND EQPDECK Entry. The ND EQPDECK entries define each Mainframe Device Interface (MDI) or Mainframe Terminal Interface (MTI) connected to the mainframe. Optionally, the ND and NT parameters are referenced in CDCNET DI configuration files. For more information about the ND entry, refer to the Deadstart Deck section of the NOS 2 Analysis Handbook.

INSTALLING CDCNET FOR THE FIRST TIME DURING A SYSTEM UPGRADE

Perform the following instructions if you are installing CDCNET for the first time as part of a NOS system upgrade or a NOS component order installation. Some of the steps are performed from the system console and some are performed from an interactive terminal.

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

1. Install CDCNET Files to User Name INSTALL.

   a. Log in to IAF under user name INSTALL. 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. Create the CDCNET installation procedure file, DCNPLIB, and load CDCNET permanent files from the permanent file tape by entering:

      SYSGEN,CDCNET.

   The permanent file tapes are requested.

d. Load the CHA1 message template file, HTF_id_psrout (where id = A, psrout = NOS release level) to disk on user name INSTALL by entering:

```
BEGIN,INITIAL,DCNPLIB,INSTALL.
```

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

e. Log out of user name INSTALL.

2. Create a NETDIR on User Name NETOPS. Perform these steps from the system console.

   a. Enter these commands:

```
X.DIS.
USER,NETOPS,password.   (supply password of UN=NETOPS)
ATTACH,DCNPLIB/UN=INSTALL.
BEGIN,INITIAL,DCNPLIB,NETOPS.
```

   b. User names NETADMN and INSTALL are given access to the directory. Once this procedure completes, enter:

```
DROP.
```

3. Load CDCNET files to user name NETADMN. Perform the following steps from an interactive terminal:

   a. Log in to IAF under user name NETADMN.

   b. Enter the following commands:

```
ATTACH,DCNPLIB/UN=INSTALL.
BEGIN,INITIAL,DCNPLIB,NETADMN.
```

   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 file names (boot files, the object library, terminal user procedures, terminal definition procedures, and the exception list) 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 user name NETADMN.

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

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

You can now use NETFM, MANCC, etc. for configuring CDCNET.

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

```
X.DIS.
USER,NETADMN,NETADMN.
BEGIN,ADDDI,DCNPLIB,type,si,pfn.
```

Parameter	Description
type	Type of DI (MDI or MTI).
si	Last six digits of the DI's system identifier. The system identifier is a 12-digit hexadecimal number that can be found inside each DI. The last 4 digits are a checksum. Do not include the checksum in this parameter.
pfn	Permanent file name for the configuration file to be created by ADDDI. This file name must be unique.

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

If you configured an MDI in step a, perform step b.

b. Execute the ADDDI procedure call again to define your TDI. Specify the DI's system identifier (si) and a unique file name (pfn):

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

After completing the procedures described in this section, refer to the CDCNET Configuration and Site Administration Guide to complete CDCNET installation.

UPGRADING CDCNET

1. Install CDCNET Files to User Name INSTALL.

   a. Log in to IAF under user name INSTALL.

   b. Make the newly released SYSGEN procedures local to your job by entering:

   ```
   GET,SYSGEN.
   ```

   c. Create the CDCNET installation procedure file, DCNPLIB, and load CDCNET permanent files from the permanent file tape by entering:

   ```
   SYSGEN,CDCNET.
   ```

   The permanent file tapes are requested.

d. Load the CHA1 message template file, HTF_id_psrout (where id = A, psrout = NOS release level) to disk on user name INSTALL by entering:

```
BEGIN,UPGRADE,DCNPLIB,INSTALL.
```

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

e. Log out of user name INSTALL.

2. Load CDCNET files to user name NETADMN. Perform the folloiwng steps from an interactive terminal:

   a. Log in to IAF under user name NETADMN.

   b. Enter the following commands:

```
ATTACH,DCNPLIB/UN=INSTALL.
BEGIN,UPGRADE,DCNPLIB,NETADMN.
```

   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 file names (boot files and object library) 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 user name NETADMN. (Configuration files, exception lists, INITDCN, terminal user procedures, and terminal definition procedures are not loaded; these files are site-controled.

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

ACTIVATING A NEW CDCNET

Activate the new level of CDCNET software using the SET_VERSION_LEVEL (SETVL) procedure in DCNPLIB. This procedure updates file INITDCN and the exception list, ELIST, to indicate the new default versions for all CDCNET DIs and the message template file. The procedure call must be performed from user name NETADMIN:

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

Parameter	Description
id	The one-character identifying letter for the CHA1 template file. The id character for the release template file is A. This parameter is required.
vvvv	The CDCNET version level. This parameter defaults to the version level of DCNPLIB.

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.

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 user name NETADMN. 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 user name NETADMN:

```
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 user name NETADMIN. By running the REFORMAT_CDCNET_LOG_FILE (REFCLF) 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 username 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 username 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:

- NETFM (Network File Manager)
- NETFS (Network File Server)
- NETLS (Network Log Server)
- NETOU (Network Operator Utility)
- NLTERM (Network Log File Termination Utility)
- NETBDF (Build CDCNET Directory File)
- NETMDF (Merge CDCNET Directory File)
- NETRDF (Restructure CDCNET Directory File)
- INITMDI (Initialize MDI)
- PIM (MDI Loader PP Program)
- NETITF (Initialize NETOU Template File)
- NAMSTRT records
- HOST message template file

CDCNET requires NAM and many of its programs and procedures.

The CDCNET network applications installation procedure modifies the network startup master files (NAMSTRT), adding new job startup calls into the network startup records (INIT, MULTI, etc.) as well as adding 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 2 Analysis Handbook.

The CDCNET Operations Handbook and the CDCNET Network Configuration and Site Administration 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 generates.

UNIQUE PARAMETERS

Parameter	Description
DEBUG	The code to aid debugging and maintenance is added. NETFS, NOF, NETLS, and NLTERM abort on certain error conditions.
TRACE	If you wish to enable AIP trace file creation for the CDCNET Host Applications and Utilities, specify the keyword TRACE on the call to the procedure for CDCNET network application installation.
ID	This parameter sets the identifying letter (A..Z,0..9) for the template file, HTF_id_psrout. The default is A. Change this letter 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 build procedure. By specifying this directive, NETFM will not use SORT5 and NETFM list output will be displayed in unsorted order. Use of this directive is not recommended.

MESSAGE TEMPLATES

When CHA1 is built, it creates a message template file; HTF_id_psrout, in addition to binaries for the deadstart tape and NAMSTRT records. In the file, id is a single character supplied to the CHA1 build procedure, and psrout 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 user name NETADMN to create a combined binary template file, TF_id_vvvv. Call the GENMSG procedure from user name NETADMN.

```
BEGIN(GENMSG,DNCPLIP,ID=id,V=vvvv,P=psrout[,PURGE])
```

Field	Description
id	The id letter of the template file created by CHA1.
vvvv	The CDCNET version level to combine with your host template file. CTF_vvvv will be ATTACHed from NETADMN.
psrout	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 will not overwrite an existing TF_id_vvvv file.

Once GENMSG is complete, you can activate the new file using the SETVL procedure described earlier in this section. 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 Special Product Installation.

CDCNET applications utilize 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(JOBLS,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, JOBLS, 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(ZZFC=500)	Flush count for log file buffer (used only by NETLS).
PARAM(OUSTART=NO)	Prevents the Network Operator Utility from starting when NAM comes up.
PARAM(LSSTART=NO)	Prevents the Network Log Server from starting when NAM comes up.
PARAM(FSSTART=NO)	Prevents the Network File Server from starting when NAM comes up.

NOTE

A YES value on any of the above three parameters causes that application to start when NAM comes up. When NOS 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 which the job intends to run. These commands have the following format:

PROG(parameter$_1$,...,parameter$_n$)

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 is 500.
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
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.

NETDIR	The network directory contains entries which match NOS permanent files to logical file references used by CDCNET Host Applications and Utilities. A default NETDIR is provided with the network.

NETLS - Network Log Server

NETLS requires the following command:

```
NETLS(MC=mc,NIN=nin,FC=fc)
```

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.
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.

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 Facility

NETOU requires the following command:

```
NETOU(NIN=nin,MC=mc,DM=dm,PROMPT,TDF=lfn)
```

Parameter	Description
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAM when the network operation is started. This parameter is required.
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if MC=0. The default value is 500. If NETOU was built without NOTRACE option, then no messages will be written.
DM=dm	Node number of the MDI; to be used as the default MDI. If you do not select a specific MDI, this MDI will be used for communicating 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.
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.

Parameter	Description
TDF=lfn	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
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 via the INITDCN procedure on user name NETADMN.
INITDCN	Procedure indicating parameter values for id and vvvv for template file name. Must be stored in user name NETADMN. This file is created as part of the CDCNET installation.

INITMDI has the following parameters:

```
INITMDI(B=lfn1,D=xx,L=lfn3,NAM=nam,EST=est,DT=dt,NIN=nin,V=ver)
```

Parameter	Description
B=lfn1	Load library local file name. If B=lfn1 is not present or B=0, the load file will be accessed using the reference in NETDIR - BOOT#ver_MCI (ver is described under the V=ver parameter that follows).
D=xx	Dump file name prefix. If D=xx is not specified or D=0, the default prefix is DI. If specified, the first character should be alphabetic, and the second should be alphanumeric. A permanent dump file will be created on the system with a name of the form xxyyNIN (xx is the prefix, yy is two alphanumeric characters that increment (AA..A9,BA..B9,etc...) for each dump taken within the same network invocation number (NIN). A reference to this dump file is created in NETDIR of the form DUMP#FULL_systemid_yymmddhhmmss (where systemid is the unique identifier for each DI and yymmddhhmmss is the year, month, day, hour, minute, and second when the dump was taken).
L=lfn3	Message history file. If L=lfn3 is not specified or L=0 is specified, no message file is produced. Messages concerning load, dump, and errors will be written to this file. PL = OUTPUT should be used.

Parameter	Description
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 will be used for error messages.
EST=est	EST ordinal of the MDI. This parameter should not be specified if automatic loading by NAM is used (NAM searches the EST table for MDIs requesting a load). If specified and if no suffix or the suffix B is used, the base will be octal. Octal range is 1 to 777. If the suffix D is specified, the value is decimal with a range 1 to 511.
DT=dt	Device type (EST name). This parameter is required and consists of two alphanumeric characters. Refer to the NOS 2 Analysis Handbook for more information. For an MDI, this parameter will be DT=ND.
NIN=nin	Network invocation number: 1- to 3-digit decimal number that will be placed in the NIN field of messages written into the L file. If not specified, the default is 000.
V=ver	Version number of the load file. This consists of hexadecimal digits.

The actual call to INITMDI is contained in the procedure INITDCN on user name NETADMN. 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,V=vvvv).

INITMDI requires the following file for defaults:

File	Description
NETDIR	This 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 existing Network Definition File in the Local Configuration File section to include application definition (APPL) statements and INCALL statements. The released NDL file, NDLDATA, on user name NETADMN contains 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.
```

For complete information on the meaning of the statements, refer to the Network Definition Language Reference Manual.

CID - CYBER INTERACTIVE DEBUG VERSION 1

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

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

COBOL5 AND COBOL5Q - COBOL VERSION 5

This 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 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 file USER 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.
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 file USER for the COBOL5 or COBOL5Q installation procedure.

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

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

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

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

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

DCL - DATA CATALOGUE VERSION 2

This section describes product dependencies and SYSGEN functions for DCL.

PRODUCT DEPENDENCIES

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

SYSGEN FUNCTIONS

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

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

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

2. Run the DCL build procedure. When the procedure is finished executing, enter these commands at the system console:

   ```
   X.DIS.
   USER(SYSTEMX,SYSTEMX)
   ATTACH(PFGDCL2/UN=INSTALL)
   SYSGEN(DCL2)
   ```

DUAL - Dual State

This section describes the following:

- Unique Parameters
- Configuration Information
- Dual State Source Library
- Dual State on Older NOS Systems
- Batch Critical Updates
- NOS and NOS/VE Memory Allocation
- NOS RUNJOBS Modification
- NOS ACCFILE
- NOS ACCFILE Modification

UNIQUE PARAMETERS

Parameter	Description
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 will be 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.

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.) At a minimum, the user name should have these additional validations added to the default NOS validations:

  AP=IAF, AP=VEIAF, AW=CASF,AW=CAND,AW=CUCP,AW=CNVE,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. 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 NOS 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 user name INSTALL then moved to user name SYSTEMX.

  If you are performing 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 user name INSTALL to load a copy of LIDCMid (and LIDLIST file) to user name INSTALL.

  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 will allow read access to the file from user name INSTALL.) 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.)

- LIDLIST file. This file defines the Logical Identifiers (LIDs) to be used by NOS/VE batch jobs. LIDLIST is an indirect access permanent file stored on user name SYSTEMX (user index 377777B). A sample LIDLIST file is provided with the release materials. This file should be edited on user name INSTALL then moved to user name SYSTEMX.

  If you are performing 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 user name INSTALL to load a copy of LIDLIST (and LIDCMid file) to user name INSTALL.

  To move the LIDLIST file to user name SYSTEMX, execute the command X.SYSGEN(MOVE,LIDLIST,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 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 Memory Allocation below.

- EQPDECK entries for sharing memory. The VEMEM utility, documented under NOS and NOS/VE Memory Allocation later in this section, will help allocate memory between NOS and NOS/VE.

- EQPDECK entries for sharing peripherals. This topic is discussed in the Deadstart Decks section of the NOS 2 Analysis Handbook.

For more information about the LIDCMid and LIDLIST files, and the LDT CMRDECK entry, refer to the LID/RHF Configuration Files section of the NOS 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 contained in file PFGDUAL. It can be obtained by executing the following command from your installation user name:

```
SYSGEN(DUAL,SOURCE)
```

This will create a file named VEDSSL which contains the Dual State Source Library in NOS/VE format. If you desire to make modifications to this file, they must be done from NOS/VE. Use these steps:

1. Log into NOS/VE and execute these commands:

   ```
   SET_LINK_ATTRIBUTES U=(INSTALL,nosfamily) PW=batchpassword
   GET_FILE $USER.DUAL_STATE_SOURCE_LIBRARY VEDSSL DC=B56
   ```

   This will get the Dual State Source Library and transfer it to NOS/VE.

2. Once changes have been made to the Dual State Source Library and you are ready to rebuild the dual-state components, move the library back to NOS. This is accomplished by the following commands:

   ```
   SET_COMMAND_LIST A=$SYSTEM.SOFTWARE_MAINTENANCE.RAF$LIBRARY
   CONVERT_DSSL_TO_TEXT $USER.DUAL_STATE_SOURCE_LIBRARY DUALpsrin
   SET_LINK_ATTRIBUTES U=(INSTALL,nosfamily) PW=batchpassword
   REPLACE_FILE DUALpsrin
   ```

   Replace psrin with the PSR level of NOS.

   This creates a file named DUALpsrin on NOS user name INSTALL. When the dual-state build procedure is executed, it will pick up this file and use it as input.

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 Software Availability Bulletin). The binaries are contained in RECLAIM dump format on file PFGDUAL. They will be loaded to disk by executing the following command from an interactive terminal logged into user name INSTALL. Remember that the RECLAIM data base must be updated and the SYSGEN procedures must be current (refer to chapter 3).

```
SYSGEN(DUAL,OLDNOS)
```

This creates files with the naming convention NVE_oldlev, 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 steps:

```
COMMON,SYSTEM.
ATTACH(NVE_oldlev)
SYSGEN(DST,SYSTEM,NVE_oldlev,NEW,0,density)
```

Replace density with the tape density of the new deadstart tape.

BATCH CRITICAL UPDATES

A Batch Critical Update (BCU) is contained on a permanent file tape. This tape contains binaries for the deadstart tape and a permanent file. The permanent file (PFGBCU2) has the same structure as the permanent file for DUAL (PFGDUAL). File one contains the Dual State Source Library, file two contains the VEMEM procedures, and file three contains a RECLAIM dump of NOS/VE binary support for back levels of NOS.

To update your deadstart tape with the corrected dual-state binaries, refer to chapter 5.

To obtain the corrected Dual State Source Library (file VEDSSL), execute the following command from an interactive terminal logged into user name INSTALL.

```
SYSGEN(DUAL,SOURCE,BCU)
```

Refer to Dual State Source Library earlier in this section for more information on how to transfer this file to NOS/VE. Note that this command will replace any existing file named VEDSSL.

To obtain the corrected dual-state binaries for back levels of NOS, execute the following command from an interactive terminal logged into user name INSTALL.

```
SYSGEN(DUAL,OLDNOS,BCU)
```

Refer to Dual State on Older NOS Systems earlier in this section for more information. Note that this command will replace any existing files named NVE_oldlev.

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.

To execute VEMEM, enter the following command from an interactive terminal logged in to user name INSTALL:

```
LINE.
BEGIN,,VEMEM.
```

VEMEM will ask 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 will display 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 2 Analysis Handbook.

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 on NOS/VE.

2. Convert the library to text.

3. Move the converted library to NOS and rebuild the library.

4. Create a new deadstart tape.

As an alternative to the above method of updating the RUNJOBS procedures, you can use the following general steps to temporarily update RUNJOBS:

1. Modify the RUNJOBS procedure in the ULIB NVELIB.

2. Place the updated NVELIB in file DSTLIB on the user name from which you invoke NOV/VE.

   Here are sample commands for temporarily updating procedure RUNJOBS:

   ```
   RETURN,SYSTEM,OLD,LGO,DSTLIB.
   COMMON,SYSTEM.
   GTR,SYSTEM,OLD,U.ULIB/NVELIB
   GTR.PROC/RUNJOBS
   ```

Edit file LGO to update the USER statement.

```
PURGE,DSTLIB/NA.
DEFINE,DSTLIB.
LIBEDIT,N=DSTLIB,U,I=0
RETURN,DSTLIB.
```

The next invocation of NVE will use the RUNJOBS procedure stored in file DSTLIB.

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:

RHACCNT — Created in the ACCFILE procedure; used for GET_FILE and REPLACE_FILE.

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.

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,&PRODUCT.
```

The following table describes the use of the partner job template. Substitution occurs in fields that are prefixed with an ampersand (&).

Field	Substitution
&JOB	The GET_FILE and REPLACE_FILE commands cause the following default command to be used: P,T5000. This job executes in the communication task service class. The CREATE_INTERSTATE_CONNECTION command uses the PARTNER_JOB_CARD parameter, if one was specified; otherwise, it uses the default job command: FASLAVE. This job executes in the batch service class. When the DUAL_STATE_ROUTE_PARAMETERS parameter is specified on the PRINT_FILE command, the following default job command is used: PRIFJOB. This job executes in the communication task service class. Interstate communications does no substitution; a NOS job command must be supplied.
&USER	The user name value specified for the SET_LINK_ATTRIBUTES command is substituted. If no SET_LINK_ATTRIBUTES command was issued, then the current NOS/VE user name is used.
&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 no SET_LINK_ATTRIBUTES command was issued, 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 no SET_LINK_ATTRIBUTES command was issued, 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 no SET_LINK_ATTRIBUTES command was issued, 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

To make changes to the ACCFILE or RUNJOBS job template, 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 the library.

4. Create a new deadstart tape.

As an alternative to the above method, you can use the following general steps to change the templates:

1. Modify the ACCFILE and/or RUNJOBS procedure in the ULIB NVELIB.

2. Place the updated NVELIB in file DSTLIB on the user name from which you invoke NOV/VE.

   Here are sample commands for updating the procedures:

   ```
   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:

   ```
   PURGE,DSTLIB/NA.
   DEFINE,DSTLIB.
   LIBEDIT,N=DSTLIB,U,I=0
   RETURN,DSTLIB.
   ```

   The next invocation of NVE will use the new procedures stored in the user library file DSTLIB.

FDBF - FORTRAN DATABASE FACILITY VERSION 1

This section describes product dependencies and unique parameters for FDBF.

PRODUCT DEPENDENCIES

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

UNIQUE PARAMETERS

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

When the SMFEX subsystem is enabled, the operating system automatically processes a large portion of the user's FSE calls through SMFEX. When SMFEX is disabled, the FSEEX program is automatically scheduled with compatible external features. Where the NOS scheduler would process the FSEEX field length of approximately 50000_8 words, the SMFEX scheduler processes approximately 3000_8 words. SMFEX can process editing transactions most effectively if the hardware configuration provides extended memory in the amount of 3000_8 words per editing user. Extended memory can be ECS, ESM, 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 54000_8 to 70000_8 words. If the configuration provides user access to extended memory, SMFEX also uses a dedicated extended field length whose size is as much extended memory as is available, up to a limit determined by installation parameter NUMSWPECS.

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

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

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

SMF PROCEDURE FILE

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

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

```
.PROC,SMFffff.
SMFEX.
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 4000_8 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 1000_8 words.

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

INSTALLATION PARAMETERS

The following parameters are defined in deck COMFSMF.

Parameter	Default	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_8 words per user. This field length can be reduced at subsystem initiation by providing less extended memory or no extended memory in the EQPDECK. NUMSWPECS should not exceed NUMSMFUSR, and need not equal NUMSMFUSR since SMFEX will use mass storage devices for overflow.

SITE-DEFINED TEACH FILE

If you want to create 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 INSTALL.

SMFSTAT PROCEDURE

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

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

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

2. Enter this command:

   ```
   BEGIN,,SMFSTAT.
   ```

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

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

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

This section describes the following:

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

MODEL PARAMETER

FTN4, FTNTS, and FTN5 reference the MODEL parameter (refer to TEXT and TEXTIO procedures in this 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 (refer to table 7-7).

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

Reinstall the compiler and CCG whenever you change parameters in OPTIONS (refer to the Build Dependency Chart in chapter 6). You can revise installation parameters in COMFCIP (called by FTN and INITOO) during the installation of FTN5, if you reassemble both FTN and INITOO.

INTEGER MULTIPLY FCOs

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

Table 7-7. MODEL Parameter and FTN4, FTNTS and FTN5 Object Code Execution

Value of MODEL Parameter in TEXT and TEXTIO	Models on Which Object Code Executes Inefficiently
71, 72, 73, 171, 172, 173, 174, 720, 730, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860	74, 175, 740, 750, 760, 176, 865, 875, 990
74	175, 176, 740, 750, 760, 990
175, 176, 740, 750, 760	74, 990
990	74, 175, 176, 740, 750, 760

Note that if the model on which the object code was compiled has LCME or UEM memory, and the compiler was built to generate LCME or UEM code, and the source program contains one or both of the following:

- Level 2 statements
- Common blocks and the AL option was used on FORTRAN compiler call

then the object code generated by such a compiler will not execute on a model that does not have LCME or UEM memory.

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

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

NOTE

The OPL deckname for the IAF procedure is IAFP.

SPECIAL NOTES

- RDFEX and IAF share the decks IAFEX and 1TM in the composite 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 file USER.

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

ITF - INTERACTIVE TRANSFER FACILITY VERSION 1

This section describes the following:

- Product Dependencies
- ITF Command
- ITF Shutdown and Idledown

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

PRODUCT DEPENDENCIES

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

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

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

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

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

$0 < PI \leq MAXACN$

$0 < CO \leq MAXTCN$

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

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) words

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

ITF SHUTDOWN AND IDLEDOWN

In the examples below, n refers to the system control point.

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

- RHF K Display command K.IDLE

- K.DISABLE,n command

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

- DROP,jsn command

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

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

To temporarily disable ITF connections, use these commands:

```
K,RHF.
K.APPL
K.DISABLE,n
```

and

```
K.ENABLE,n
```

Where n is the ordinal assigned to ITF.

To terminate ITF operation, use these commands:

```
K,NAM.
K.AP=NFV
K.DISABLE,AP=ITF
```

and

```
K.IDLE,AP=ITF
```

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

```
K,RHF.
K.PATH
K.DISABLE,n
```

and

```
K.ENABLE,n
```

Where n is the ordinal assigned to the LID and/or PID.

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 file USER in the installation job for LCS.

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

Table sizes and central memory requirements are shown in table 7-8.

Table 7-8. Table Sizes and Central Memory Requirements

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

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

*DEFINE CBLCOPY

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

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

*DEFINE COLUMN6

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

LOADER - CYBER LOADER VERSION 1

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

UNIQUE PARAMETERS

PRESET — The PRESET parameter sets the default central memory presetting options. The default 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 control statement, it will override this default setting.

Value	Preset (Octal)
NONE	No presetting for ECS; same as ZERO for CM
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
NFINF	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 of OFF. If the MAP parameter is specified on the LDSET control statement or the MAP control statement 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

FLMSG — This parameter determines if LOADER issures a dayfile message indicating the field length required for loading and execution. The default is YES.

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 FOR LOADER

You can change the following parameters for Loader. Insert the parameter changes at LDRCOM.13 in the 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 100_8.
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. 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 file is rewound prior to beginning of load; one of the following values. Value — Description 1 — File is rewound. 0 — File is not rewound.

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

MAP SUBSYSTEM

There are four DECKOPL installation procedures for installing the MAP subsystem: MSSI, 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
MSSI - MAP subsystem	BLDMLIB, MSAMLIB
MMCL - Macro library control	None
AP1I - Online diagnostics	MEMSIZE
AP1L - Online diagnostics	MAPTYPE, ADD, MUL, DIV, SQRT

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

MSSI VALIDATION REQUIREMENTS

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

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

```
AP1UN=username
AP1PW=password
```

To change the user name or password, replace the parameter values username and password with the new user name and password.

PROCEDURES FOR INITIATING MSSI

CDC provides three procedure files for initiating MSSI:

MAPCMI for initiating MAP IV-23/25

MAPCH for initiating MAP IV-20/21

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

INSTALLATION PARAMETERS

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

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 the beginning of this chapter for information about the MCS procedure file and about subsystem initiation.

NOTE

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

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

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

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

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

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

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

Consider the following two procedures.

Example 1:

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

Example 2:

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

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

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

SPECIAL NOTES

- To activate debug dumps for the ADL processor, include a *DEFINE DEBUG directive in file USER for the MCS installation job.
- Users must be validated to access MCS (refer to MODVAL in the NOS 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	If you want to save M86LIB as a direct access file, specify the keyword SAVELIB on the call to install MSE.

CONFIGURATION INFORMATION

To configure the Mass Storage Extended Subsystem (MSE) 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 SUBFAMO (user index 377760B). A sample MSECONF file is provided with the release materials. This file should be edited on user name INSTALL then moved to user name SUBFAMO.

  If you are performing a first time installation, ATTACH the example file from user name SUBFAMO directly. If you are installing MSE for the first time during an Upgrade or Component Order installation, execute the SYSGEN(MSE1) procedure from user name INSTALL to load a copy of MSECONF, MSE, MSESLAV files to user name INSTALL.

  To move the MSECONF file to user name SUBFAMO, execute the command SYSGEN(MOVE,MSECONF,SUBFAMO,,Y,PERMIT) from the system console when directed to by instructions in chapters 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 multi-mainframe environment (file name MSESLAV). The MSE startup procedures are indirect access files stored on user name SYSTEMX (user index 377777B). These files should be edited on user name INSTALL then moved to user name SYSTEMX.

  If you are performing 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 user name INSTALL to load a copy of MSE, MSESLAV, and MSECONF file to user name INSTALL.

  To move the either file to user name SYSTEMX, execute the command SYSGEN(MOVE,filename,SYSTEMX,,Y,PERMIT) from the system console when directed to by instructions in chapters 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 Archival Subsystem and Deadstart Decks sections in the NOS 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.
MAXSMUNIT	2	Number of storage modules allowed in the Unit Device Table. Maximum is 8.
MAXSMM1	1	Must equal MAXSMUNIT - 1.

COMXIPR Parameters

COMXIPR contains the following parameters used by MSE executive (SSEXEC). Assemble CALLFAS to obtain a listing of COMXIPR.

Parameter	Default	Description
NUMSLV	3	The number of slave mainframes for which master SSEXEC can service file staging requests.
NUMRB	9	The number of file staging request blocks available to a slave mainframe.
SLRP$INTV	5	The number of seconds SSEXEC waits to look for new staging requests from slave mainframes.
SLAV$INTV	60	The number of seconds SSEXEC waits with no signal from a slave mainframe before assuming that the slave executive has terminated.

MSS - MASS STORAGE SUBSYSTEM VERSION 1

This section describes the following installation options for MSS:

- Unique Parameters
- MSS Procedure File
- Special Notes
- Common Deck Parameters

UNIQUE PARAMETERS

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

MSS PROCEDURE FILE

Refer to Subsystem Initiation, at the beginning of this chapter, for information about the MSS procedure file and about subsystem initiation. SYSGEN(FULL) installs a sample MSS procedure file named MSS on user name SYSTEMX.

SPECIAL NOTES

The common deck COMUSIT contains parameters that affect MSS. Refer to the NOS section of this chapter for descriptions of the common deck COMUSIT parameters.

COMEIPR Parameters

COMEIPR contains the following parameters used by the MSS executive (MSSEXEC). Assemble CALLMSS to obtain a listing of COMEIPR.

Parameter	Default	Description
NUMSLV	2	The number of slave mainframes for which the master MSSEXEC can service file staging requests.
NUMRB	9	The number of file staging request blocks available to a slave mainframe.
SLRP$INTV	5	The number of seconds that MSSEXEC waits to look for new staging requests from the slave mainframes.
SLAV$INTV	60	The number of seconds that MSSEXEC waits with no signal from a slave mainframe before assuming that the slave executive has terminated.

NAM5/NAM5D NETWORK ACCESS METHOD VERSION 1

This section describes the following installation options for NAM:

- Configuration Information
- 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. The NCFFILE and LCFFILE reside as direct access permanent files on user name NETADMN. These files must be public, semiprivate, or private files with read permission given to user name NETOPS.

  If you are installing a NAM/CCP network, the NDL must contain both an NCF and an LCF. If you are installing a NAM/CDCNET network, only the LCF is required. More details on the two portions of the NDL are given below.

- Network Configuration File (NCFFILE). The NCF portion of an NDL 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 that 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 defines the network applications, which 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 user name NETADMN. This file defines a simple one-NPU network which includes the definitions for the PSU printer and all CDCNET applications. This file may be examined by logging in to user name NETADMN or by executing SYSGEN(NDL1) from user name INSTALL to load NDLDATA (and the compiled version of NDLDATA, NCFFILE and LCFFILE) to disk.

To compile an NDL, use NDLP as described below:

1. Log in to user name NETADMN 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=INSTALL.
   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,NETOPS=R.
   PERMIT,LCFTEMP,NETOPS=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.
   ```

- NP EQPDECK entry. This entry defines the connection of each 255X NPU to your mainframe. The ND parameter on this statement is referenced by the NODE parameter of the NDL COUPLER statement.

- ND EQPDECK entry. 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 2 Analysis Handbook for more information about the NP and ND EQPDECK entries. Refer to the CDCNET Configuration and Site Administration Guide for more information about CDCNET configuration files.

SPECIAL NOTES

- The NAM installation procedure installs the following NAM components and utilities:

  NAMI (Network Initialization Program)
  NIP (Network Interface Program)
  PIP (Peripheral Interface Program)
  NS (Network Supervisor)
  CS (Communications Supervisor)
  NVF (Network Validation Facility)
  DLFP (Debug Log File Processor)
  AIP (Application Interface Program)
  QTRM (Queued Terminal Record Manager)
  TVF (Terminal Verification Facility)
  NDLP (Network Definition Language Processor)
  LFG (CCP Load File Generator)
  NDA (NPU Dump Analyzer)
  COLLECT (Collect Network Dumps)
  LISTPPM (Format PIP Memory Dumps)

- NAM5 binaries are released on the deadstart tape in non-debug/trace mode. CDC 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 procedure and NAMSTRT files from the program library on NAM5. The installation procedure saves these files as public indirect access permanent files on the installation user name. The SYSGEN utility automatically moves the files to SYSTEMX and NETOPS. Other optional product installation procedures modify NAMSTRT (PSU, RBF, PTF/QTF with NAM interface, CDCNET, and ITF). Thus, you should not move NAMSTRT until these optional products have been installed.

- The NAM1 Host Application Programming Reference Manual describes NIP, PIP, NS, CS, AIP, QTRM, and DLFP. The NAM/CCP Terminal Interfaces Reference Manual describes TVF. The Network Definition Language Reference Manual describes NDLP. The NOS 2 Analysis Handbook describes LFG, NAMI, COLLECT, and NDA.

- There are execution time interdependencies between NAM and CCP. These interdependencies are established in file USERBPS (refer to CROSS and CCP 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 JOBNS, JOBCS, JOBTVF, JOBNIP, and JOBNVF save the log files as direct access permanent files on tape and purges them. For network problems, this tape (with other support materials) should be included with all PSRs submitted for network products. A more detailed description of the log file capability is in the NAM1 Host Application Programming Reference Manual.

UNIQUE PARAMETERS

Parameter	Description
NOTRACE	If you want to disable trace/log file creation for NS, NVF, TVF, and CS, 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 2 Analysis Handbook for a description of the NAMI command and for further details on NAMNOGO and the CFO command.

The released startup master file, NAMSTRT, is a multirecord file containing six parameter records (INIT, RESTRT, RECOVR, MINIT, MULTI, and MRECOV) and seven job records (JOBNIP, JOBNS, JOBCS, JOBNVF, JOBTVF, JOBCOL, and JOBPUR). For the initial NAM startup, you should specify the parameter record INIT for a single host network, or MINIT if your network contains more than one host. Subsequent entries of NAM from the console cause NAMI to use the parameter record RESTRT. If you have a multihost network, you should change the NAMI command in the NAM procedure file so that subsequent entries from the console use the parameter record MULTI.

The INIT record directs NAMI to route to the input queue jobs to start the programs NS, CS, NVF, TVF, and COLLECT. NIP is started at the NAMI control point.

Based on JOB and PARAM statements in the INIT record, other network applications (RBF, ITF, PSU, PTF/QTF, CDCNET Host Applications) 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 file USER for the NAM5 installation procedure.

NOTE

You should be thoroughly familiar with NAM operations before defining DEBUG and/or STAT. DEBUG and STAT are defined by the release defaults. To remove the definitions, specify NOTRACE on the call to the NAM5 installation procedure.

name	Significance When Defined
DEBUG	Code to aid in debugging and maintenance in NIP and PIP is generated. The following shows the effect of DEBUG on NAM components. • NIP, CS, NS, and NVF abort on certain error conditions. • CS, NS, and NVF are loaded with the debug version of AIP and produce network traces. • PIP hangs PPs for certain error conditions. • NIP uses internal trace buffers to trace messages sent and received and to trace subroutine and overlay calls.
IMS	Descriptive internal maintenance comments are included in the assembly and compilation listings.
STAT	Additional statistics-producing code is generated in NIP and AIP. With STAT defined, each time an application stops talking to the network, a terminal-to-application connection terminates, or an application-to-application connection terminates, statistical information is written to the NIP dayfile. After NIP terminates, the dayfile indicates the number of times each overlay was called and gives the statistics kept in common block STATTAB. The size of the job dayfile increases significantly when STAT is defined.
ZZDN	Code is generated to log all inbound or outbound messages between PIP and NIP in local file ZZZZZDN.

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 user name INSTALL by SYSGEN. You can modify the file using a text editor or through Update directives against the NAM PL.

A comment can follow the parameters on all statements and directives.

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 that you can use to describe the purpose of the record.

Example:

```
TITLE(PARAM) INIT - INITIAL NETWORK INVOCATION
```

Parameter Records

Parameter records contain two kinds of directives that tell NAMI what parameters to substitute in job skeletons (PARAM directives) and what jobs to start (JOB directives). Each record can consist of a number of lines or directives up to 80 characters long, terminated by a zero byte. Refer to the NOS 2 Analysis Handbook for a description of parameter records available with the released system.

Refer to figure 7-8 for an example of a parameter record.

```
INIT TITLE(PARAM) INIT - INITIAL NETWORK INVOCATION.
.*
.*
.*  THIS PARAMETER RECORD IS SELECTED WHEN THE NETWORK
.*    IS TO BE STARTED WITH FRESH LOADS OF THE FIRST
.*    LEVEL NPU(S), AND THEIR PREVIOUS CONTENTS
.*    ARE NOT TO BE DUMPED.
.*
.*  1.  PURGE ALL PREVIOUS NETWORK DUMPS AND TRACES.
.*
.*  2.  LOCAL NPU(S) WILL BE STOPPED AND RELOADED
.*      WHEN THE NETWORK IS STARTED.
.*
.*  3.  LOCAL NPU(S) WILL NOT BE DUMPED WHEN THEY
.*      ARE INITIALLY LOADED.
.*
.*  4.  LOCAL NPU(S) WILL BE STOPPED IF THE HOST
.*      NETWORK FAILS.
.*
.*  5.  A FAILING NPU WILL TRIGGER HOST SUPERVISOR
.*      PROGRAM FIELD LENGTH DUMPS TO BE TAKEN AND
.*      PREVIOUS TRACE FILES TO BE PRESERVED.
.*
.*
PARAM(NCFFN=NCFFILE)          NETWORK CONFIGURATION FILE.
PARAM(LCFFN=LCFFILE)          LOCAL   CONFIGURATION FILE.
PARAM(NLFFN=NLFFILE)          NETWORK LOAD FILE (CCP).
PARAM(NETUN2=NETADMN)         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

PARAM Directive

The PARAM directive is used in the parameter record to define keywords and values that are substituted for matching keywords in the job skeleton records. PARAM has the following format.

PARAM($keyword_1$=$value_1$,$keyword_2$=$value_2$,...$keyword_n$=$value_n$) comment

Multiple PARAM directives can appear in a parameter record.

The following rules apply to the PARAM directive:

- Embedded spaces are ignored.
- Keywords and values can contain only letters, digits, and asterisks.
- Keywords and values must be no longer than 7 characters.
- If a keyword appears more than once, only the first definition applies.

Example:

When the following directive is present, every occurrence of the keyword NCFFN in any job skeleton record is replaced by the string NCFFILE.

```
PARAM(NCFFN=NCFFILE)  NETWORK CONFIGURATION FILE.
```

JOB Directive

The JOB directive specifies the name of a job skeleton record and a code for the name of the network product that the job skeleton starts. A JOB directive must appear in every parameter record for each Network Host Products program that NAMI should start. The JOB directive has the following format:

JOB(name,type[,ssname,at_1,at_2,at_3]) comment

Parameter	Description
name	Name of job skeleton record.
type	First two or more characters of an application or program name, such as NS, CS, NIP, and so on. (Only the first two characters are used.)
ssname	Subsystem name if program is a subsystem (not required for NIP).
at_i	Job attribute. Any of the following are valid:

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.
SY	If specified, the job record is routed with system origin privileges.
NS	If specified, the job record is routed with the Network Supervisor service class (SCL=NS).

The rules for format of a PARAM directive apply to the JOB directive.

Example:

```
JOB(JOBNIP,NIP)  NAM (NIP) JOB.
```

Job Skeleton Records

Each job skeleton record contains the commands and input records required to start one network program. The record is in NOS job file format. In any of the commands or input record lines, any keyword (1 to 7 letters, digits, or asterisks, delimited by separators) may be a substitutable parameter. That is, if any keyword in the job skeleton record is identical to a keyword in the parameter record, NAMI substitutes the corresponding value from the parameter record for the keyword in the job skeleton. If a separator is an underscore, NAMI removes the underscore from the record when it makes the replacement.

In addition to substituting values for keywords defined in the parameter record, NAMI also substitutes variables known to it at the time it executes. These variables pertain to the startup master file currently in use and to the names of dump and trace files which NAMI creates with each new startup of NIP. Certain reserved keywords have been defined for use in the job skeleton records wherever one of the NAMI variables should be substituted.

Keyword	Meaning
CIN	Current network invocation number.
OIN	Old network invocation number.
UNM	Master file user name.
PWM	Password for master file user name.
MFN	Master file name.

The following keywords are defined by NAMI and should be used in the job skeleton record wherever the dump and trace files are to be referenced.

Keyword	Meaning
xxD0FIL	Program dumps.
xxD1FIL	
xxD2FIL	Binary dumps of field length.
xxD3FIL	
xxL0FIL	Listable output.
xxT0FIL	ZZZZZDN.
xxS0FIL	ZZZZZSN.

xx is the first 2 characters of the type from the JOB directive in the parameter record.

Job skeleton records must be constructed so that the files whose existence is assumed by the various programs are present.

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 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    NIDOFIL   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    NILOFIL   JOB DAYFILE.
.*      ZZZZZDN   TRCLEV1             NIP TRACE FILE WRITTEN BY NIP.
.*                TRCLEV2   ZZNIFIL   INTERMEDIATE NIP TRACE FILE.
.*                TRCLEV3   NITOFIL   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.
```

Figure 7-9. Example of Job Skeleton Record (Sheet 1 of 5)

```
.*
PURGE(ZZNI_OIN,ZZNI_CIN/NA)
.*
.*  START NIP.
.*
RFL(60000)
NIP(NIN=CIN,ISTP=NIISTP,FSTP=NIFSTP,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(NITOFIL,ZZNI_CIN/NA)
RETURN(OUTPUT)
WRITER(OUTPUT)
EXIT.  NIP
.*
.*  SAVE RUN DATA IF AVAILABLE.
.*
ENDIF(SAVFILS)
.*
IF(FILE(OUTPUT,AS),NOUTPUT)
ATTACH(NIPOUT=NIDOFIL/NA,M=W)
IF(.NOT.FILE(NIPOUT,AS))      DEFINE(NIPOUT=NIDOFIL)
REWIND(OUTPUT)
SKIPEI(NIPOUT)
COPYEI(OUTPUT,NIPOUT)
RETURN(OUTPUT,NIPOUT)
ENDIF(NOUTPUT)
.*
IF(FILE(ZZZZZPP,AS),NOZZZPP)
ATTACH(PIPDMP=NID1FIL/NA,M=W)
IF(.NOT.FILE(PIPDMP,AS))      DEFINE(PIPDMP=NID1FIL)
REWIND(ZZZZZPP)
SKIPEI(PIPDMP)
COPYEI(ZZZZZPP,PIPDMP)
RETURN(ZZZZZPP,PIPDMP)
ENDIF(NOZZZPP)
.*
```

Figure 7-9. Example of Job Skeleton Record (Sheet 2 of 5)

```
IF(FILE(ZZZZDMB,AS),NOZZDMB)
ATTACH(NIPDMB=NID2FIL/NA,M=W)
IF(.NOT.FILE(NIPDMB,AS))       DEFINE(NIPDMB=NID2FIL)
REWIND(ZZZZDMB)
SKIPEI(NIPDMB)
COPYEI(ZZZZDMB,NIPDMB)
RETURN(ZZZZDMB,NIPDMB)
ENDIF(NOZZDMB)
.*
IF(FILE(ZZZZTMP,AS),NOZZTMP)
ATTACH(NIPZTMP=NID3FIL/NA,M=W)
IF(.NOT.FILE(NIPZTMP,AS))      DEFINE(NIPZTMP=NID3FIL)
REWIND(ZZZZTMP)
SKIPEI(NIPZTMP)
COPYEI(ZZZZTMP,NIPZTMP)
RETURN(ZZZZTMP,NIPZTMP)
ENDIF(NOZZTMP)
.*
ATTACH(TRCLEV2=ZZNI_CIN/NA)
IF(FILE(TRCLEV2,AS),NTRCLV2)
SKIPR(TRCLEV2)
IF(.NOT.FILE(TRCLEV2,EOF))     REWIND(TRCLEV2)
ELSE(NTRCLV2)
IF(FILE(ZZZZZDN,AS),NOTRACE)
ENDIF(NTRCLV2)
ATTACH(TRCLEV3=NITOFIL/NA,M=W)
IF(.NOT.FILE(TRCLEV3,AS))      DEFINE(TRCLEV3=NITOFIL)
SKIPEI(TRCLEV3)
COPYEI(TRCLEV2,TRCLEV3)
PURGE(ZZNI_CIN/NA)
IF(FILE(ZZZZZDN,AS),NTRCLV1)
RENAME(TRCLEV1=ZZZZZDN)
REWIND(TRCLEV1)
IF(ZZMC.NE.0)                  SKIPR(TRCLEV1)
COPYBF(TRCLEV1,TRCLEV3)
BKSP(TRCLEV3)
SKIPR(TRCLEV3)
IF(.NOT.FILE(TRCLEV3,EOF))     WRITEF(TRCLEV3)
ENDIF(NTRCLV1)
RETURN(TRCLEV1,TRCLEV2,TRCLEV3)
ENDIF(NOTRACE)
.*
ATTACH(NIPLST=NILOFIL/NA,M=W)
IF(.NOT.FILE(NIPLST,AS))       DEFINE(NIPLST=NILOFIL)
SKIPEI(NIPLST)
NOTE(DFL,NR)/NIDA_CIN
DAYFILE(DFL)
PACK(DFL)
COPYEI(DFL,NIPLST)
.*
RETURN(OUTPUT)
WRITER(OUTPUT)
EXIT.  NIP
.EOR
```

Figure 7-9. Example of Job Skeleton Record (Sheet 3 of 5)

```
.*
.*
.*  THIS JOB IS SUBMITTED EVERY ZZMC MESSAGES TO PLACE
.*    THE TRACE INFORMATION FROM THE PROGRAM (LEVEL 1) ONTO
.*    THE INTERMEDIATE PERMANENT FILE ZZNIFIL (LEVEL 2).
.*
.*  IF ALL THAT HAPPENS IS THAT THIS JOB IS REPEATEDLY
.*  SUBMITTED THEN THE TRACE INFORMATION IS KEPT FOR
.*  ONLY THE LAST 2 TIMES ZZMC MESSAGES.
.*
.*  THIS CONSTRAINS THE SIZE OF THE TRACE FILE KEPT
.*  WHEN THE NETWORK IS RUNNING WITHOUT ANY PROBLEMS.
.*
.*
NIPA_CIN,T77777.     DUMP AIP TRACE TO ZZNIFIL.
USER(UNM,PWM)
ATTACH(TRCLEV2=ZZNI_CIN/M=W,NA)
IF(FILE(TRCLEV2,AS),NTRCLV2)
SKIPF(TRCLEV2)
COPYBF(TRCLEV2,TEMP)
REWIND(TRCLEV2,TEMP)
COPYBF(TEMP,TRCLEV2)
ELSE(NTRCLV2)
DEFINE(TRCLEV2=ZZNI_CIN)
WRITEF(TRCLEV2)
ENDIF(NTRCLV2)
COPYBF(INPUT,TRCLEV2)
BKSP(TRCLEV2)
SKIPR(TRCLEV2)
IF(.NOT.FILE(TRCLEV2,EOF))    WRITEF(TRCLEV2)
SETJOB(DC=NO)
EXIT.  NIPA
.EOR
.*
.*
.*  THIS JOB IS SUBMITTED IN RESPONSE TO A
.*    *HOP RELEASE DEBUG LOGFILE* COMMAND.
.*
.*  THE PURPOSE OF THIS JOB IS TO SAVE ON THE LEVEL 3 PERMANENT
.*  FILE *NITOFIL* THE PREVIOUS 2 TIMES ZZMC MESSAGES
.*  CURRENTLY IN THE INTERMEDIATE (LEVEL 2) FILE *ZZNIFIL*
.*  BEFORE WRITING THE NEW TRACE DATA (FROM LEVEL 1 FILE)
.*  ON THE INTERMEDIATE (LEVEL 2) FILE. THIS WILL ALLOW THESE
.*  TRACE MESSAGES TO BE COLLECTED AND WRITTEN TO TAPE.
.*
.*
NIPB_CIN,T77777.     DUMP TO PERMANENT TRACE FILE.
USER(UNM,PWM)
ATTACH(TRCLEV2=ZZNI_CIN/M=W,NA)
IF(FILE(TRCLEV2,AS),NTRCLV2)
SKIPR(TRCLEV2)
IF(.NOT.FILE(TRCLEV2,EOF))    REWIND(TRCLEV2)
ATTACH(TRCLEV3=NITOFIL/NA,M=W)
IF(.NOT.FILE(TRCLEV3,AS))     DEFINE(TRCLEV3=NITOFIL)
```

Figure 7-9. Example of Job Skeleton Record (Sheet 4 of 5)

```
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 (Sheet 5 of 5)

Network Host Product (NHP) Program Requirements

Job skeleton records for the CS, NIP, NS, and NVF jobs must each contain a command that calls the program that the job intends to run. These commands have the following format:

prog(keyword$_1$=value$_1$,...keyword$_n$=value$_n$)

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
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.

Parameter	Description
CP=cputil	CPU utilization threshold for NPUs, from 50 through 100. The default is 100. If CPU utilization exceeds this value, the NPU operator is alerted.
BU=bufutil	Buffer utilization threshold for NPUs, from 0 through 500. The default is 0. If available buffers drop below this level, the NPU operator is notified.

CS may not be required in every host of a multihost network. If you do not want CS, remove the CS job statments from the network startup master file.

CS requires the following files:

Name	Description
NCF	Network configuration file created by NDLP. The permanent file name is NCFFILE. Residence of the NCFFILE is controlled by the NETUN2 PARAM statement in the NAMSTRT parameter records. The released default is NETADMN.
NRF1	Job record to be copied to the ZZZZZDN file by NETREL, that determines the disposition of the network trace file, when NETREL is called on a periodic basis.
NRF2	Job record to be copied to the ZZZZZDN file by NETREL, that determines the disposition of the network trace file, the next time 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
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAMI when the network operation is started. This parameter is required.
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if mc is 0. The default value is 0.
ISTP=istp	Option to stop all local NPUs during network host initialization. The default is NO. **Value** — **Description** YES — Stop all local NPUs. NO — Do not stop local NPUs.
FSTP=fstp	Option to stop all local NPUs upon network host termination. The default is YES. **Value** — **Description** YES — Stop all local NPUs. NO — Do not stop local NPUs.
INACT=ito	Inactive timeout. Time in minutes for connection to be inactive. Default is 10 minutes. After such period NIP will send FC/INACT/SM to application. If the value is zero, no FC/INACT/SM will be sent. The range is 0 to 99.
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 4. The minimum value is 2.
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 8. 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 4. The minimum value is 2.
MAXFL=maxfl	Maximum field length that NIP can reach. The range is from 61440 to 12280. The default is 98304.

Use the following formula to determine a value for MAXFL for a particular configuration. (Round MAXFL to the nearest multiple of 64.)

$$\text{MAXFL} = 11096 + 700h + 6560a + 30(c+d) + 30e + m + 20\ (k_1w_1 + \ldots\ k_nw_n) + 22y + 13z + 78b_1 + 142b_2 + 206b_3$$

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.
k_1w_1	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:

k_1 Application block limit (ABL) and downline block limit (DBL). Use one of the following values:

Value	Description
1	ABL $\leq$ 2 and DBL $\leq$ 2.
ABL - 1	ABL > 2 and DBL $\leq$ 2.
DBL - 1	ABL $\leq$ 2 and DBL > 2.
ABL + DBL - 2	ABL > 2 and DBL > 2.

w_1 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
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAMI when the network operation is started. This parameter is required.
FDP=fdp	Forced dump option. The default is NO.

Value	Description
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.
NO	NPUs are not dumped in the first 10 minutes of program execution except for the initial load.

Parameter	Description
RT=rt	Release trace file option. The default is YES.

Value	Description
YES	NAM requests NS programs CS and NVF to release their trace files whenever NS dumps an NPU.
NO	Trace files are not requested to be released.

Parameter	Description
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if mc is 0. The default value is 500.
NDFCT=option	File category of NPU dump files. The following options can be specified.

Option	Description
PU	Public file.
S	Semiprivate file.
P	Private file.

NS may not be required to run in every host of a multihost network. If you do not want NS, remove the NS job statements from the network startup master file.

NS requires the following files:

Name	Description
NLF	Network load file created by LFG in the CCPLOAD procedure (refer to the NOS 2 Analysis Handbook for a description of LFG). The permanent file name is NLFFILE. Residence of the NCFFILE is controlled by the NETUN2 PARAM statement in the NAMSTRT parameter records. 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.

Parameter	Description
NIN=nin	Network invocation number; 1- to 3-digit decimal number assigned by NAMI when the network operation is started. This parameter is required.

NVF requires the following files:

Name	Description
LCF	The local configuration file created by NDLP. The permanent file name is LCFFILE. Residence of the LCFFILE is controlled by the NETUN2 PARAM statement in the NAMSTRT parameter records. The released default for NETUN2 is NETADMN.
NRF1 and NRF2	Described under CS, earlier in this section.

NOS AND NOS2B - NETWORK OPERATING SYSTEM

The NOS procedure assembles all COMPASS routines; the NOS2B procedure compiles the FORTRAN and SYMPL routines.

This section describes the following:

- Configuration Information
- Installation Parameters
- 819 PPU Driver Installation

CONFIGURATION INFORMATION

NOS is configured by making entries into several text records, known as the 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 Parameters 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 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. Table 2-3 contains an index of the 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 line numbers, which you need to change an installation parameter.
- Put the NOS installation deck parameter changes on file USER and again execute the MODOPL procedure.

If you change any of the installation parameters in a NOS deck, reassemble all routines that use that deck. Use the KRONREF command to determine which routines use the NOS deck.

Refer to table 7-9 for brief descriptions of the NOS common decks that contain installation parameters.

Table 7-9. 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.
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.
COMUSIT	CALLMSS	ASMOVE utility parameters.
PPCOM	PPTEXT	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 2 Administration Handbook for further information on VALIDUS.
AUFN	VALINDS	Micro definition that specifies the name of the available user indexes file. Refer to the NOS 2 Administration Handbook for further information on VALINDS.

The NOS 2 Analysis 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 2 Reference Set, Volume 3).
APXT	7777_8	Default user password expiration term in days, from 1 through 7777_8. This value is assumed when MODVAL sets a password for a new user name. APXT must be less than or equal to APXL. A value of 7777_8 indicates the password cannot expire.
KTLI	10B	Value used in calculating the default time limit; the maximum value is 176_8. For details of the algorithm used in the calculation, refer to the NOS 2 Analysis Handbook.
KLPI	1000B	Default limit for lines printed from a file; the maximum value is 3776_8.
KCPI	0	Default limit for cards punched from a file; the maximum value is 76_8.
KMSI	1000B	Default limit for additionally allocated mass storage PRUs; the maximum value is 7776_8.
KDFI	100B	Default limit for dayfile messages written; the maximum value is 176_8.
KCCI	100B	Default limit for commands processed; the maximum value is 176_8.
KECI	0	Default limit for extended memory field length/1000_8; the maximum value is 176_8.
KCMI	37B	Default limit for central memory field length/100_8; the maximum value is 37_8.

Parameter	Default	Description
KSLI	10B	Default limit for SRU accumulation; the maximum value is 76_8.
KDTI	0	Default limit for the number of detached jobs; the maximum value is 37_8.
KPTI	1000B	Default limit for the number of units plotted; the maximum value is 76000_8.

COMSBIO Parameters

COMSBIO contains the following parameters, which are used for control of BIO functions. Assemble CALLSYS to obtain a listing of COMSBIO.

Parameter	Default	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
PR$	Defined	Any line printer.
LR$	Defined	Any 580-12 printer.
LS$	Defined	Any 580-16 printer.
LT$	Defined	Any 580-20 printer.

Parameter	Default	Description
SB$	Defined	Punch system binary.
PB$	Defined	Punch system binary.
P8$	Defined	Punch 80-column binary.
PU$	Defined	Punch coded.
PH$	Defined	Punch coded.
PL$	Defined	Plotter.
LX$	Defined	Any 5870 nonimpact printer.
LY$	Defined	Any 5970 nonimpact printer.
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	If you did not initialize a mass storage device during deadstart (using the INITIALIZE entry described in the NOS 2 Analysis Handbook), the system searches the device for a label that might be in track 0. This parameter specifies the number of tracks the system searches before determining that the device has a bad label or no label. When it reaches that track number, it stops searching for a label. If the device is a system device, the system writes a new label; if it is not a system device, the error codes LE (label error) and U (unavailable) status are entered in the mass storage table (MST), and the device must be initialized after deadstart. MST is described in the NOS 2 Analysis Handbook.

COMSMLS Parameters

COMSMLS contains micros that define security access levels and access categories. Redefining any of the access level or access category micros requires reassembly of all programs that reference them. A site security administrator supplies any changes to be made to these micros. Assemble CALLSYS to obtain a listing of 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.

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
MXSY	5	Maximum number of devices that can be defined as system devices during deadstart.
AFDL	20000B	Account file threshold size in PRUs.†
BLTL	1000B	Binary maintenance log threshold size in PRUs.†
DFDL	20000B	Dayfile threshold size in PRUs.†
ELDL	1000B	Error log threshold size in PRUs.†
UPTL	10B	Maximum number of uncorrected processor errors that can occur per hour before the operator is notified.
SYSCHG	SYSTEM	Specifies the system change number for jobs initiated under DSD.
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.

†When entries in any of these files reach the threshold, the A,OPERATOR flashing message appears on the console screen (refer to the NOS 2 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.
NTIM	300	Delay time in seconds (decimal) before reissuing the CHECK, E,P DISPLAY message at MAGNET´s control point when entries exist in MAGNET´s preview buffers.
POLM	0	Flag indicating whether all tape hardware error messages are issued to the job dayfile. If POLM is 0, the system issues only the first and last messages to the job dayfile. If POLM is 1, the system issues all tape error messages to the dayfile. The user can override the installation setting of POLM with parameters on the tape assignment command (refer to the NOS 2 Reference Set, Volume 3). The system issues all tape error messages to the error log regardless of the setting of POLM.
POGH	0	Flag indicating whether the system allows hardware-detected correctable errors when writing on 6250-cpi group-encoded (GE) tapes. The user can override the installation setting of POGH with parameters on the tape assignment command (refer to the NOS 2 Reference Set, Volume 3). If POGH is 0, the tape subsystem performs write error correction according to industry standard group-coded recording (GCR) techniques. Control Data recommends this setting because it provides efficient throughput, error recovery, and tape use when writing GE tapes on media suitable for use at 1600 cpi or 6250 cpi. If POGH is 1, hardware GCR error correction is disabled. Control Data recommends this option only for special archiving and diagnostic applications. Successful use requires higher-than-normal quality tape and special drive adjustments. Use in a normal environment generally results in increased error rates, decreased throughput, and decreased tape capacity. Use only tape that is suitable for recording at 6250 cpi when this setting of POGH is in effect. Because use of the disabled GCR error correction mode (also known as perfect write) may necessitate additional maintenance activities, consult site maintenance personnel before making this the default mode of operation.

Parameter	Default	Description
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: 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: 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 will be timed-out and aborted.

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 DEFT 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 a site security administrator.
FPXT	7777_8	Default permanent file password or permission term in days, from 1 through 7777_8. This value is assumed when the user does not specify an expiration term parameter on a permanent file request. FPXT must be less than or equal to FPXL. A value of 7777_8 indicates the password or permission cannot expire. Changes to this parameter should be supplied by a site security administrator.
MNHS	5	Minimum size hole, in sectors, that permanent file manager (PFM) creates in the indirect access file chain when using an existing hole. If, in the search for a hole in which to save an indirect access file, PFM finds that the use of an existing hole creates a new hole containing fewer sectors than MNHS, then PFM allocates space at the end of the indirect access chain. If a delink operation creates a hole smaller than MNHS, PFM delinks one less track to ensure minimum size for the hole. The purging of a file whose total length is less than MNHS results in the creation of a hole smaller than MNHS. If a value for MNHS is smaller than the average length of the indirect access files on the system, it results in holes that may be unusable. If the value is larger than the average file length, it results in holes which are not used for a period of time. For efficient use of holes, the value for MNHS should be close to the average length of the indirect access files on the system. The minimum value for MNHS is 3; the maximum value is 7777_8.

Parameter	Default	Description
BRDE	BRAL	Backup requirement (BR) default specifications; can be set to the following symbolic values. **Value** — **Description** BRAL — Backup always required. BRMD — Media-dependent backup for systems with a supplemental mass storage facility. BRNO — No backup required. GRMX — Maximum BR value.
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.
PMLM	62_{10}	Number of explicit permissions allowed per file, 1 through 4095. If PMLM is changed, PFM must be reassembled.
PGUI	300000_8	PFDUMP (refer to the NOS 2 Analysis Handbook) purge limit user index. When PFDUMP is used with the purge option, selected files under user indexes greater than PGUI are dumped, but not purged. If PGUI is changed, PFDUMP must be reassembled.

For individual users, each of four permanent file access limits is established through MODVAL (refer to the NOS 2 Administration Handbook) by specifying a range index from 0 through 7. Each range index corresponds to an upper limit specified by one of the following installation parameters. The last character of the installation parameter indicates the range index being defined. Table 7-10 summarizes the released values for each parameter. Setting a parameter to 0 indicates unlimited access.

Parameter	Description
NFRNGn	Upper limit of range n for file count; must not exceed 77777_8.
CSRNGn	Upper limit of range n for cumulative size of indirect access files, specified in PRUs; must not exceed 777777_8.
FSRNGn	Upper limit of range n for size of individual indirect access files, specified in PRUs; must not exceed 77777_8.
DSRNGn	Upper limit of range n for size of individual direct access files, specified in PRUs; must not exceed 777777_8.

Table 7-10. Released Values of Permanent File Limit Ranges

Parameter	Values of n†						
	1	2	3	4	5	6	7
NFRNGn	10	20	30	40	50	100	0
CSRNGn	1000	2000	5000	10000	50000	100000	0
FSRNGn	10	30	50	100	150	300	0
DSRNGn	1000	2000	5000	10000	50000	100000	0

†All values are specified in octal; 0 indicates unlimited access.

COMSPRO Parameters

The following COMSPRO parameters contain a general description of the PROFILa file. Assemble CALLSYS to obtain a listing of COMSPRO.

Parameter	Default	Description
PPFN	PROFILC	Micro definition specifying the PROFILE routine's database file name (refer to the NOS 2 Administration Handbook).
PPWD	SECURUS	Micro definition specifying the PROFILE routine's database file password.
PUSN	SYSTEMX	Micro definition specifying the catalog location of the PROFILE routine's database.

COMSREM Parameters

COMSREM contains the following parameters, which are used by the Interactive Facility (IAF) executive. Assemble CALLSYS to obtain a listing of COMSREM.

Parameter	Default	Description
UTIS	10D	Value used in calculating the default CPU time limit in seconds for any particular terminal job's activity, if it is not specified with the SETTL command (refer to the NOS 2 Reference Set, Volume 3). For details of the algorithm used in calculating the default time limit, refer to the NOS 2 Administration Handbook.
VDSI VDTI	100B 100B	Default system resource unit (SRU) and time limit increment values for the S,nnnnn and T,nnnnn interactive commands.
VXLL	2500D	Maximum number of characters in a logical input line.
VXPH	2500D	Determines the physical line length that IAF accepts. IAF uses VXPH to calculate a buffer length.
TAPC	20B	Number of pots of typed ahead input to be stored in IAF's FL for each user.
VNCP	40B	Number of pots of output to be stored in IAF's FL for each user.
NMFL	60000B	Defines the size of NAM's field length as used by the algorithm in COMPCMX. This calculation determines the field length available for an interactive job. This value should be equal to the value of MAXFL, which defines the maximum field length that NAM can attain. MAXFL is a parameter on the NIP command.

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
RPMS	4	Length of time that a job waiting for an auxiliary device is rolled out before retrying assignment.
RPOV	8	Length of time that a job that has had a request for an auxiliary device denied because of overcommitment deadlocks is rolled out before retrying assignment.
SUBM	10	If MAGNET is not active, length of time that a noninteractive service class job calling RESEX is rolled out before retrying assignment.
MTMS	2	When one of the following situations prevents immediate assignment of the tape, MTMS specifies the length of time that a job requesting a tape is rolled out before retrying the assignment. • The job requests a tape with a VSN that is not currently available. • The job requests a 9-track tape that is mounted without a write ring, the job requests the wrong tape density, and the tape hardware detects that the density of the tape is incompatible with the unit on which it is mounted (800-cpi tape on a 1600/6250-cpi drive or 6250-cpi tape on an 800/1600-cpi drive).
RFTL	10	Length of time that a job requesting a resource is rolled out before retrying the request when a track limit occurs on the resource demand or VSN files.
MTOV	8	Length of time that a job that has had a request for a magnetic tape denied because of overcommitment deadlocks is rolled out before retrying the assignment.

COMSSCD Parameters

COMSSCD contains the definitions of the service classes used by NOS. Assemble CALLSYS to obtain a listing of COMSSCD. By default, four installation service classes are defined: I0, I1, I2, and I3. To define additional installation service classes, add CLASS macro calls similar to those already present. You can also change the attributes of the defined installation service classes (those that are parameters on the CLASS macro). However, if you make any changes, reassemble all the decks that call COMSSCD.

COMSSFS Parameters

COMSSFS parameters are used by the MODVAL and PROFILE commands. Assemble CALLSYS to get a listing of COMSSFS.

Parameter	Default	Description
FLLM	50000_8	Specifies the field length limit for the execution of the MODVAL and PROFILE commands. If the execution of a MODVAL or PROFILE command requires more than the specified field length, disk storage is used. Accessing disk storage is more time-consuming than accessing central memory and degrades performance.

COMSSRU Parameters

COMSSRU contains the parameters used in SRU calculations. Assemble CALLSYS to obtain a listing of COMSSRU. Refer to COMSSRU in the NOS 2 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
MINP	4	Defines the number of pool PPs reserved for use by NOS.
MINDP	MINP/2	Defines the number of driver PPs reserved for use by NOS on a C180-81/830 with 20 PPs.

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 build 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 2 Administration Handbook). Assemble CALLTAB to obtain a listing of COMTNAP. This common deck does not contain any executable code. Each table entry has the following format.

59	17	11 0
application name (6-bit display code, left-justified, blank-filled)	reserved	application validation word bit position

Bits 17 through 12 of each entry are reserved for the program that uses COMTNAP. These bits are set to 0 when COMTNAP is assembled. The last word of the table must be 0.

Each application defined in COMTNAP must appear only once. However, any application validation bit can appear more than once; that is, a given application validation bit can be defined to permit use of more than one application, if the operations at a particular site make such a definition desirable. Bits 47 through 36 of the application validation word are reserved for customer application use; bits 35 through 15 are reserved for Control Data application use.

You need to modify COMTNAP only for applications that allow terminal-to-application connections. Do not define applications that use NAM only for application-to-application connections.

The released table defined by COMTNAP follows:

Application Name	Bit Position
IAF	0
RBF	1
TAF	2
MCS	3
TVF	4
CS	5
PNI	6
ITF	7
TLF	8
NJF	9
NETOU	10
PSU	11
AP1	12
AP2	13
AP3	14

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	17_8	Table length, excluding zero-word terminator.

COMUSIT Parameters

COMUSIT contains the following parameters, which are used by the ASMOVE utility to control selection of files for destaging to the mass storage facility (MSF) and for releasing their disk space. Compile CALLMSS to obtain a listing of COMUSIT. Compile ASMOVE to determine how the following weight and scale factors are used in the selection algorithm.

Parameter	Default	Description
DEFDB	2	Weight factor for MSF-preferred residence.
DB$SCALE	1.0	Scale factor for MSF-preferred residence.
DEFDC	1	Weight factor for no preferred residence.
DC$SCALE	1.0	Scale factor for no preferred residence.
DEFDL	1	Weight factor for file length.
DL$SCALE	1.0	Scale factor for file length.
DEFDT	0	Weight factor for time since last modification.
DT$SCALE	1.0	Scale factor for time since last modification.
DEFDV	24	Weight factor for destage control value.
DV$SCALE	25.0	Scale factor for destage control value.
DEFMN	6	Weight factor for minimum file length.
MN$SCALE	25	Scale factor for minimum file length.
DEFMX	128	Weight factor for maximum file length.
MX$SCALE	250	Scale factor for maximum file length.

PPCOM Parameters

PPCOM contains the following parameters, which are used by system peripheral processor packages for intercommunication. Assemble PPTEXT to obtain a listing of PPCOM.

Parameter	Default	Description
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.
UEBS	1000_8	Number of words in a block of user ECS (1000_8, 2000_8, 4000_8).
ESMX	512D	Maximum number of EST entries, plus one. The value for ESMX can range from 10 through 512.
LCOM	12_8	Maximum length of the L-display input buffer in words. The value for LCOM can range from 1 through 12_8.

Parameter	Default	Description
LDSY	350_8	Maximum length of the L-display output buffer in words. The value for LDSY can range from 100_8 through 1000_8.
MSMX	200D	Maximum number of mass storage devices. The value for MSMX can range from 1 to 200.

The assembly constant INSP$ is defined in NOSTEXT. If the INSP$ reference is deleted, 10 bytes in both the DSD display and the command overlays are unavailable for site code.

DSD Parameters

The following parameters, specified in ENTER macro calls (within the DSD syntax tables), cause the first 25 characters of the associated DSD command to be logged in the system dayfile and/or the error log. The commands are logged just as they are entered by the operator except that the characters

```
DS,
```

are placed before each command. The DSD listing contains an explanation of the ENTER macro. Assemble DSD to obtain a listing.

Parameter	Default	Description
SDF	-	When specified in an ENTER macro call, the associated command is logged in the system dayfile.
ERL	-	When specified in an ENTER macro call, the associated command is logged in the error log. On the release tapes, the OFF, ON, channel control, and memory commands specify ERL on their ENTER macro calls.

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
- 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 has been completed. The station has been tested with the following validations, some of which may not be necessary.

MT=7	DF=77B
PR=7	SL=77B
SC=50B	AW=CLPF, CSPF, CCNR, CASF, CAND, CSRP, CSAP
MS=77B	CS
CM=20B	CC
TL=77B	CA
CP=77B	CA
DB=7	CS
LP=77B	CS
CC=77B	

  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 build procedure. (File ZZSYSGU should also be updated to reflect the new username/password).

- 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, Remote Host Facility (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-state section of this chapter. If you are not installing dual state, create the LIDCMid file on user name INSTALL 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 user name INSTALL.) 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 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 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.

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 will transfer more than one file. SOT76 is initiated when communication is established between mainframes and does not terminate until the station is dropped. All other SPOTs are initiated as required to transfer one file, and each is terminated after completion.

Field length requirements fluctuate as file transfers are initiated and terminated. As file transfers are terminated, buffers are released. Definitions in the common deck COMTUNE control the lengths of the buffers used by the SPOTs. Each SPOT uses one of the COMTUNE SYMPL DEFs to determine the length of the I/O buffers. All SPOTs are written in the SYMPL language.

Nominal release definition octal values for the I/O buffers are described in the following list.

Name	Value	Description
LBUFDSTP	3004	Deadstart via tape
LBUFDSLM	7725	Deadstart link buffer
LBUFDP	4001	7000 dump
LBUFLM	1401	Link medium
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 7-11.

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 7-11. SPOT Octal Memory Requirements

SPOT Type	Code	Buffer Lengths Used to Transfer Files	Total Memory Used	Notes†
Spooling	10200	LBUFSP+LBUFLM	14600	Single file transfer
Permanent file	5100	LBUFPF76+LBUFLM	11600	
Tape staging††	6700	3*MBL+LBUFLM/2+3	11100	For MBL<(LBUFTP-LBUFLM/2-3)/3 [7690]
		LBUFTP	14700	For (LBUFTP-LBUFLM/2-3)/3 [7690] <MBL<LBUFTP-LBUFLM-1 [23090]
		MBL+LBUFLM+1	20500	For MBL>LBUFTP-LBUFLM-1 [23090]
Dump	4200	LBUFDP	10200	
Deadstart	6200	LBUFDSTP+LBUFDSLM	21200	Via tape

†Figures in brackets are the results using the released default values for the parameter variables in the equations.
††The buffer length required for tape staging depends on the value specified for the maximum block length (MBL) for tape staging operations. Use the equations in the Notes column to determine the appropriate values to use. The maximum value that can be specified for MBL is 50000.

INSTALLATION PARAMETERS

The following installation parameters are all on common deck COMTUNE on the NOS station PL. Deck COMTUNE also lists the minimum, maximum, and default values for these parameters.

Table Size Parameters

Parameter	Description
NMF	The number of mainframes to which the station can be linked; the released value for NMF is 1, and the maximum value that can be specified is 2.
DATAENT	The maximum number of active data streams.
SNTS	The size of the SPOT name table (SNT). The size of the SNT should be the value of DATAENT plus the maximum number of local SPOTs (four) plus the spooling SPOT (one).

Parameter	Description
IDTMAX	The maximum size of the IDT table controls the number of logical IDs a mainframe can have and allows this value, minus two logical IDs. IDTMAX must match IDTL defined in PP program SSH; otherwise, SSH hangs the spooling SPOT until the station is dropped.
MAXSPOTS	A group of parameters that define the default maximum number of active SPOTs of each type that MFSTAT activates at one time.
SPOOLS	The maximum number of spooling streams.
DISNAMESIZE	The size of the display name table for the transparent display interface to the SCOPE 2 operating system.
SYNTAXSIZE	The size of the syntax extension table for the transparent display interface to the SCOPE 2 operating system.

Spooling and Recalling Time Parameters

Parameter	Description
MSEC(i)	Recall time in milliseconds.
SEC(i)	A method of approximating the number of busy station loops in i seconds.
ISEC(i)	A way of approximating the number of idle station loops in i seconds.
TME7000	The delay in seconds between sending time requests to the SCOPE 2 operating system.
STA7000	The delay in seconds used by MFSTAT between sending status requests to the SCOPE 2 operating system.
RCL7000	The recall time in milliseconds used by the busy overlay of MFSTAT when communicating with a linked SCOPE 2 operating system.
TIMEOUT	The length of time in seconds that MFSTAT waits before logging out a linked mainframe when communication is lost.
MSGCNT	The length of time in seconds that MFSTAT leaves informative messages on the B display.
BSYLIM	The length of time in seconds the busy overlay of MFSTAT delays after sensing an idle condition, before going into an idle state.
IDLRCLTM	The recall time in seconds used by MFSTAT when the idle overlay is executing.

NOTE

For better response time, lower both the RCL and STA values. To reduce CPU use, increase the RCL value. If the RCL and STA parameters are reduced too much, however, the reduction may cause STD (the link medium coupler driver) to be locked in.

Parameter	Description
LOOPLIM	The delay in seconds that MFSTAT waits before checking for a change in busy to idle status (controls the frequency with which the busy portion of the station checks its busy status).
DSDWAIT	The length of time in seconds that MFSTAT waits for a reply before it rejects a DSD request.
MAXINCOUNT	The frequency in seconds with which MFSTAT checks the input queues for files to spool when it is idle.
OVLMAX	The maximum time in seconds that MFSTAT retains the secondary overlay field length after a load of a secondary overlay.
IDLEMAX	The elapsed time in seconds that MFSTAT waits after all spooling activity has completed before swapping out the spooling SPOT.
SPLLIM	The number of seconds MFSTAT waits before going idle once spooling activity is completed.
IDLETIME	The number of seconds the spooling SPOT waits before trying to initiate spooling operations.
IDLETIME2	The amount of time in seconds the spooling SPOT waits before initiating new spooling operations if output spooling is taking place.
SPOOLRCL	The recall time in milliseconds used by the spooling SPOT when there is no spooling activity.

Buffer Size Parameters

Parameter	Description
DAYBUFSIZE	The size of the MFSTAT buffer for processing SPOT dayfiles on the active side.
RRBUF	The size of the MFSTAT active transmit buffer and the linked staged packet buffer.
LRGBUF	The size of the MFSTAT receive buffer for the active side of the station.
LICRBUF	The length of the MFSTAT buffer used by the IAF queue utility helper.

Parameter	Description
LBUFLM	The length of the link buffer used by SPOT jobs for NOS-SCOPE 2 I/O spooling.
LBUFSP	The length of the disk buffer used by the spooling SPOT for NOS-SCOPE 2 spooling of I/O files.
LBUFPF76	The length of the disk buffer used to read and write the disk for permanent file transfers to and from SCOPE 2.

The following installation parameter is on the deck HELLO7:

Parameter	Description
NOMFF	If NOMFF is set to 1 (the release value), HELLO7 supports a single mainframe. If multiple mainframe support is desired, NOMFF must be set to 0.

PSU - PRINTER SUPPORT UTILITY VERSION 1.1

PSU is released in binary format only. However, there are a few special considerations for installing PSU binaries. This section describes the following:

- Configuration Information
- PSU Command
- Special Versions of PSU
- PSULINK Procedure
- File Placement
- NDL File Entries

CONFIGURATION INFORMATION

To configure the Printer Support Utility, create the following:

- NDL file entries (NAM/CCP network only). If your asynchronous printer is connected to a 255X NPU, 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.

  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 NCFFILE and LCFFILE 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. If your asynchronous printer is connected to a CDCNET MTI or TDI, entries are required in several CDCNET configuration files. A DEFINE_LINE statement must appear in the system configuration file for the DI to define the connection between the DI and the printer. In addition, a terminal user procedure (TUP) and a terminal definition procedure (TDP) are required for proper configuration of the printer.

  Sample CDCNET configuration files are provided with the release materials. Refer to the CDCNET Configuration and Site Administration Guide for a description of these files.

- Electronic Vertical Format Unit Load File (EVFULFN). This file sets default options for PSU and is stored as an indirect access permanent file on user name NETOPS. The file contains legible text and is normally modified using PSU directly, but may be modified in advance using the instructions below.

  If you are performing a first-time installation, GET EVFULFN from user name NETOPS 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 user name INSTALL to load a copy of EVFULFN and the PSU NAMSTRT records (file NAMSTRT is required) to user name INSTALL.

  To move the EVFULFN file to user name NETOPS, execute the following command from the system console:

  ```
  SYSGEN(MOVE,EVFULFN,NETOPS,,Y,PERMIT)
  ```

  Refer to the instructions in chapters 2, 3, or 4. (The PERMIT parameter allows read access to the file from user name INSTALL.)

For more information about PSU, consult the PSU section in the NOS 2 Analysis Handbook.

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.

SPECIAL VERSIONS OF PSU

PSU is released in Notrace/64-character set format. To install a Trace 64-character set version, use the SYSGEN(SWAP) function described in chapter 8.

Save the released Notrace 64-character set version of PSU if you think you may need it later--the Notrace 64-character set version is released only on the deadstart tape. You can, however, recreate a Notrace version of PSU by using the PSULINK procedure.

PSULINK PROCEDURE

The PSULINK procedure in DECKOPL allows you to link PSU relocatables. Use PSULINK to do the following:

- Apply critical corrections to PSU. A critical correction to PSU consists of a file containing relocatables of PSU routines. These relocatables must be placed in the direct access file PSUREL on user name INSTALL before executing the PSULINK procedure. This applies to both 63- and 64-character set sites.

- Create a 63-character set version of PSU. File 3 of file PFGPSU1 on the released permanent file tapes contains relocatables for PSU. These relocatables are automatically loaded from tape when the PSULINK procedure is executed. If file PSUREL exists on user name INSTALL, the permanent file tapes will not be used.

By default, the PSULINK procedure produces a TRACE version of PSU. To generate a NOTRACE version of PSU, specify the keyword NOTRACE on the PSULINK procedure call.

Run the PSULINK procedure after the NAM5 and FTN5 procedures have completed or after the SEED procedure has been run with the REBUILD option. When PSULINK has completed, the PSU binaries are located in file PRODUCT.

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 into your installation user name to add the PSU record to NAMSTRT. SYSGEN(PSU1) requests your released permanent file tape and loads the file PFGPSU1. This file contains the released NAMSTRT records and a sample EVFULFN. The EVFULFN file must be installed to user name NETOPS. For more information about the EVFULFN file, refer to the NOS 2 Analysis Handbook.

You can move both the NAMSTRT and EVFULFN files to the appropriate user names by using the SYSGEN(MOVE) commands (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) of the following format:

  linex: LINE,PORT=portx,LTYPE=A2,TIPTYPE=ASYNC,LSPEED=9600.

- A TERMDEV statement in the NCF of the following format:

  devicex: TERMDEV,TC=M33,AUTOCON,HN=hostnode,LK=YES,OC=YES,PA=0,PW=0,ABL=3,DBL=2.

- A USER statement in the Local Configuration File (LCF) of the following format:

  devicex: USER,MFAM=family,MUSER=PRINT0n,MAPPL=PSU.

Parameter	Description
linex	Line name for a PSU printer.
portx	Port number for a PSU printer line.
devicex	Device name for a PSU printer.
hostnode	Node number of the host coupler to which the PSU printer is connected.
family	Name of the family containing user name PRINTOn where n is a number from 1 to 8. User name PRINTOn must be validated to use PSU.

The released NDL file (located on user name NETADMN), NDLDATA contains the definition for two PSU printers.

QU3 - QUERY UPDATE VERSION 3

The installation tool SYNGEN, and the various common decks that reside on the DDL 3 program library, must be available for the installation of Query Update 3.

The QU3 interface to the Database Utilities, version 1, for CRM logging is provided by releasing the DBU relocatable binaries needed to load the QU/CRM logging capsule, CAPLOG, on the permanent file tapes. SYSGEN(SOURCE) loads this file to the installation user name INSTALL. The QU3 installation deck contains commands to access this file and to create a temporary library, DBULIB, to be used at load time. The deck also contains commands to add the command-callable portion of DBU (DFRCV) to the PRODUCT file, thus allowing inclusion of DFRCV in the deadstart tape. No special definitions are required to access this interface.

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. CDC 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 procedure. (This option is not available for RBF5D.)

RBF COMMAND

RBF5 requires the following command.

RBF2PO(MC=mc)

Parameter	Description
MC=mc	Maximum message count; specifies the maximum number of messages to be accumulated in the debug log file before NETREL is called. No NETREL call is issued if mc is 0. This parameter is optional; no NETREL call is issued if MC is omitted. If NETREL is to be called, a file NRF1 must be assigned to the control point before the command is executed. File NRF1 must contain a valid job record for writing to the ZZZZZDN file. The released RBF job skeleton record creates NRF1 from the job input record.

USER FILE DIRECTIVES

To assemble various features into RBF, include directives of the form:

*DEFINE name

in file USER for the RBF5 installation job.

name	Significance When Defined
DEBUG	Code to aid in debugging and maintenance is generated.
IMS	Descriptive internal maintenance comments are included in the assembly and compilation listings.
TRACE	Symbolic table dumps of RBF are written to file SPITOUT when RBF fails.

The following parameters are defined in the common deck IP$COM. To make changes to these parameters, place appropriate Update directives on file USER for the RBF5 installation job.

Parameter	Default (decimal)	Description
SEARCHTIME	15	Time interval in seconds between scans of the output queue for remote batch files. These times are increased by approximately 10 seconds when the load on RBF is light and when most of RBF's field length is rolled out to disk.
RESUMETIME	20	Time interval in seconds between receipt of the last interactive message and the automatic switching of the terminal to batch mode; should be larger than SEARCHTIME.
REFRESHTIME	30	Refresh period in seconds for the RBF console queue displays when RBF is specified on the DISPLAY command; should be larger than RESUMETIME.
STATIONS	16	Maximum number of consoles.
TOTDEV	32	Maximum number of batch devices.
MAXFL		Maximum field length to which RBF expands when obtaining buffers. If TRACE is defined, the default value is 100000; if TRACE is not defined, the default value is 50000.
ALOTIME	600	Time in seconds that a dial-in terminal is allowed to remain inactive before being timed out of RBF. A value of 0 specifies that terminals are not timed out of RBF. The maximum value allowed is 4095.

FILE PLACEMENT

Execution of the RBF5 installation procedure requires a prior build of NAM and the existence of the NAM startup master file under the build user index. The RBF5 installation procedure modifies the startup master file (NAMSTRT). Because other products may require the startup master file, refer to NAM5 File Placement in this section to determine when and where to move the file.

RDFEX - REMOTE DIAGNOSTIC FACILITY

RDFEX and IAF share decks IAFEX and 1TM in the composite 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 user name INSTALL then moved to user name SYSTEMX.

  To create the RCFMid file, first make a file containing network configuration statements on user name INSTALL. This file is then processed by the system utility RCFGEN to produce RCFMid. An example sequence of commands is given below:

  1. Create RCFGEN 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 RCFGEN), access the new version from GLOBLIB:

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

  3. Execute RCFGEN:

     ```
     RETURN,RCFMid.
     PURGE,RCFMid.  (Be sure to PURGE any previous file.)
     DEFINE,RCFMid.
     REPLACE,rcfin.
     RCFGEN,I=rcfin,L=list,O=RCFMid.
     RETURN,RCFMid.
     ```

     File list will contain any diagnostic messages from RCFGEN.

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 two-character alphanumeric identifier from the MID CMRDECK. 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 state section of this chapter. If you are not installing dual state, create the LIDCMid file on user name INSTALL 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 user name INSTALL.) 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. (CDLT 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 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 do, that file must contain a RETURN,RHF command to return the local file RHF before the call to RHF. For example,

```
.PROC,RHFffff,...
RETURN,RHF.
local site commands
RHF.
```

RHFffff can be a procedure file stored in an indirect access permanent file under the system user index (377777_8).

Refer to the beginning of this 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 NAD Hardware Reference Manual for further information.

INSTALLATION PARAMETERS

Parameter	Description
MAXFILEXFR	The maximum number of file transfers that the Facility Interface Program (FIP) allows for any one application. Values may range from 1 through 10. Default is 4. The current definition format is:

```
1     7             22
      DEF MAXFILEXFR  #4#;
```

The change deletion location is in common deck COMADEF. The following is an example of a parameter change.

```
DEF MAXFILEXFR  #5#;
```

FETBUFSIZE	The number of words assigned to buffer space for each file transfer. Values may be zero or greater, but FIP overrides low values. The current definition format is as follows:

```
1     7             22
      DEF FETBUFSIZE  #3200#;
```

FETBUFSIZE	Assigned (binary)	Assigned (coded)
0 to 532	532	992
532 to 992	532 to 992	992
992 and up	992 and up	992 and up

The default value for FETBUFSIZE is 3200, which corresponds to about 49 PRUs. Values larger than 6400 (98 PRUs) do not increase transfer rates appreciably, and make job swapping more likely because of the increased central memory required.

The change deletion location is in common deck COMADEF. The following is an example of a parameter change:

```
DEF FETBUFSIZE #2800#;
```

NAD Controlware Initialization Parameters

A set of initialization parameters must be loaded into NAD memory along with the NAD controlware as documented in the 380-170 NAD Hardware Reference Manual. These parameters are compiled into MHF and are appended to all NAD controlware loads. The default values provide for a maximum of 25 remote NADs, 24 paths, use of all available NAD memory, and no NAD buffer tracing. To change any of the initialization parameters, you must modify the default values and reinstall 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 controlware 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 section of the NOS 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. However, you must update the deadstart tape to contain the microcode.

To build NAD microcode, submit the following job. The binaries produced by this job are copied to permanent file PRODUCT.

```
job command.
UER,username,password,familyname.
LABEL,TAPE,F=fmt,tapetyp,D=density,LB=k.
NOTE,IN.+*COMMENT PPU/170 170 Firmware M6401-xxx(PNyyyyyyyy)
COPYBF,INPUT,INHOLD.
BEGIN,170,INSTALL.
---eoi---
```

Parameter	Description
fmt	Format of tap. For example I or SI.
tapetyp	Tape Type. For example, MT or NT.
density	Tape density. For example, PE or 6E.
xxx	Version level. For example, D05.
yyyyyyyy	Part number. For example, 12345678.

RHP - PTF/QTF FILE TRANSFER FACILITY

This section describes the following options for RHP:

- Configuration Information
- Unique Parameters
- Installation Parameters
- QTF Initialization Procedure Parameters
- QTF Configuration Directives

CONFIGURATION INFORMATION

To configure the Remote Host Products (PTF/QTF), create the following:

- RCFMid file. Entries in this file allow PTF/QTF file trnasfers to take place using an RHF LCN network. The entries define the PTF/QTF applications (APPL statements) and the inter- and intra-host 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, consult the NOS dual state section of this chapter. If you are not installing dual state, create the LIDCMid file on user name INSTALL using an available text editor.

  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 will allow read access to the file from user name INSTALL.) 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. (CDLT 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 which 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 definition of the network products gateway to CDCNET using the DEFINE_NP_GW command in the configuration file for the Mainframe Device Interface (MDI) which contains the X.25 line.

A sample NDL file containing PTF/QTF application definitions is provided with the release materials. To examine this file, consult 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 will contain 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 needed to 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 section of the NOS 2 Analysis Handbook. Additional information about the LDT CMRDECK entry can be found in the Deadstart Decks section of that manual. Information about the NDL entries can be found in the Network Definition Language Reference Manual.

Information about the CCP NLFFILE as well as sample configurations which use PTF/QTF can be found in the CCP section of chapter 7 of this handbook. Information about the DEFINE_NP_GW command can be found in the CDCNET Configuration and Site Administration 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. Here are the values you can specify for subsys: RHF NAM BOTH The default is SUBSYS=RHF.

Parameter	Description
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 7-12 shows the binaries that are built depending on which subsystem you have specified.

Table 7-12. 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 queue file transfer facility (QTF) can have active at any one time. Values can range from 1 through 10; the default is 4. (Lower values reduce QTF´s memory requirements, but may also reduce the number of queue files transferred simultaneously.) The current definition (acceptable to both COMPASS and SYMPL) is as follows:
	
	`1         11    18   24        36`
	`#ACNMAXC  #DEF#  4    #ACNMAXC  #4#;`
	
	The change deletion location is in common deck COMCAPR. An example of a parameter change follows.
	
	`#ACNMAXC  #DEF#  2    #ACNMAXC  #2#;`

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 seconds. Default is 600. The current definition format (acceptable to both COMPASS and SYMPL) is as follows:
	
	`1         11    18   24        34`
	`#TIMEOUT  #DEF#  600D #TIMEOUT  #600#;`
	
	The change deletion location is in common deck COMCAPR. The following is an example of a parameter change.
	
	`#TIMEOUT  #DEF#  400D #TIMEOUT  #400#;`

MAXRTRY	The number of retries that an application attempts to successfully complete a file transfer 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#  20   #MAXRTRY  #20#;`

QTF INITIATION PROCEDURE PARAMETERS

The QTF initiation procedure has two parameters. For a full description of these parameters, refer to the NOS 2 Analysis Handbook.

QTF CONFIGURATION DIRECTIVES

Refer to the NOS 2 Analysis Handbook for information about modifying QTF configuration directives.

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	You can install TAF with or without the trace feature. To get the trace feature, specify DEBUG on the call to the installation procedure TAF.
TASKLB	To create a task library (refer to TASK LIBRARY below), specify the keyword TASKLB on the call to TAF installation procedure.

TASK LIBRARY

Before running the TAF installation procedure, create a task library permanent file containing the following required tasks:

Task	Description
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.

Task	Description
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 TAF procedure file and about 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 names and passwords for running TAF--user name KB100DC and password TAFPASS under user index 16 octal. The user name is used by SYSGEN to install the released TASKLIB file and for TAF operations. To change the password, supply a USER directive in the TCF file. To change the user name or user index, reassemble TAF changing the USNM and TRUI micros. These micros are set in deck COMKIPR.

INSTALLATION PARAMETERS

Unless otherwise specified, the following parameters are defined in deck COMKIPR. These parameters specify the charge and project numbers and user index for TAF. They also specify the user name under which TAF runs.

Parameter	Default	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 the 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_8. ECSFL cannot be less than 0 nor greater than 400_8.
NCMB	40	Actual number of communication blocks allowed in the subsystem. Communication blocks hold incoming terminal input. This parameter can be changed by the initialization command K.CMB, but it cannot be less than 19 nor greater than 40.
NSCP	31	Maximum number of subcontrol points. It cannot be less than 2 nor greater than 31.
SCMFL	376600B	Maximum field length. SCMFL cannot be less than 40000_8 nor greater than 376600_8, and must be set to a value that can be attained.
TLFM	TASKLIB	Micro whose string specifies the system task library file name.

The following parameters, defined in deck TAF, specify the default DSDUMP parameters. The user can override the parameters specified on CMDUMP requests with a task.

Parameter	Default	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.

The following parameters specify default time dependencies. Although these values are expressed in milliseconds, they are accurate to only 1 second.

Parameter	Default	Description
CORTL	1*1000	How often TAF checks to see if memory can be released to the system.
ITRTL	1500	Time to wait for input before rollout of transaction executive field length. ITRTL is defined in deck TAF.
RRTTL	1*1000	Time allowed to elapse before evicting a reusable task.
TACTL	2*60*1000	Time allowed to elapse between TAF receiving any input and TAF generating a call to ITASK. TACTL is defined in deck TAF.
TROTL	10*60*1000	Duration of rollout. TROTL is defined in deck TAF.
DMMTL	4	Time allowed to elapse between calls to the data manager(s).
TSKTL	120	Task time slice in milliseconds.

The following parameters, defined in deck TAF, specify default task rollout parameters.

Parameter	Default	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.

The following 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.
IPTAR	1	Automatic recovery flag:

Value	Description
0	Automatic recovery is disabled.
1	Automatic recovery is enabled.

If recovery is disabled, the following requests are not honored in recovery mode.

Request	Comments
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.

Parameter	Default	Description
RECDF	0	Default user recovery flag: Value — Description 0 — User recovery is enabled. 1 — User recovery is disabled.
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.
MAXJL	2500	Maximum word count on one journal request to any journal file, including header words; defined in deck TAF.
MAXRA	500B	Task limit for RA+1 requests; defined in deck TAF.
MAXTO	6*MAXWS	Maximum number of words task can send to the communication subsystem. Reaching or exceeding this value causes the task to abort.
MAXWS	409+1	Number of words SEND can transmit plus 1. Exceeding this value causes the task to abort.
TLDL	TLDLE*10	Amount of space to reserve for added tasks in the TAF-resident copy of the directory of each task library attached by TAF. This space can be used when TAF is informed of a task library change through the LIBTASK TT option. The value of the symbol should be a multiple of the size of a task library directory entry (TLDLE, currently 3). The default value allows space for 10 (TLDL/TLDLE) additional tasks. If more than TLDL/TLDLE tasks are added by the TT option, only the first TLDL/TLDLE tasks can be executed. The next time TAF is reinitialized, however, all the tasks added via the TT option are available to be executed. TLDL is defined in deck COMKTLD.

The following parameters are used with the TAF/CRM Data Manager.

Parameter	Default	Description
AOBFL	31	Output queue length.
AIBFL	31	Input queue length.
CMAXDB	31	Number of CRM databases.

Parameter	Default	Description
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 appendix G of 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.
BMAX	8	Number of before-image recovery files. The maximum value for BMAX is 63.
RMDM	1	Number of mainframes running TAF/CRM Data Manager.

The following parameters are used with TOTAL Data Manager and are defined in deck TAF. For information on the installation of TOTAL, refer to the NOS 2 Applications Installation Handbook.

Parameter	Default	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 parameters are 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.
NCTL	250	Maximum number of terminals in network communication table (NCT). To reduce core storage requirements, NCTL may be less than the total number of terminals in the network file (each entry requires 3 CM words). NCTL should be greater than or equal to the maximum number of terminals logged in at one time. If NCTL is exceeded, a terminal is rejected upon login. If the number of terminals defined in the NCTFi file is less than NCTL, the number of terminals in NCTFi replaces the value specified by NCTL. NCTL is defined in COMKIPR.

NOTE

The installation parameter MAXRA must be equal to or greater than the value for NCTL for successful initialization of TAF. This is due to the processing that CTASK must complete for every user during initialization.

The following parameters specify the default communication block parameters.

Parameter	Default	Description
CBDL	57	Length of the data input area in the communication blocks. This parameter is in deck COMKCBD.
CBUL	9	Length of user area in the communication blocks. This parameter is in deck COMKCBD.
NCBC	4	Maximum number of communication blocks reserved for large transaction input.
NLIN	4	Maximum number of users allowed to perform large transaction input simultaneously. TAF reserves n - NLIN * 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
TLDMT	600	Total number of tasks in all libraries. This includes all system tasks and TLDMN free slots for each task library directory. The maximum value for TLDMN is 1365 - (4 times number of user task libraries).
TLDMN	10	Number of tasks that may be added online to TAF's copy of any particular task library directory by the LIBTASK TT option.
TRDMN	10	Number of transactions that may be added online to TAF's copy of any particular transaction library directory by the LIBTASK TT option.
TRDMT	300	Number of named transactions per library.

The following parameters, defined in deck DMREC (except where otherwise indicated), specify the default batch recovery parameters.

Parameter	Default	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 — Seven-track tapes. 1 — Nine-track tapes.
EXPCT	10	Default value of the percentage parameter for the EXPAND directive of deck DMREC.
FTABL	5000	Length of intermediate ignore table used during DMREC recovery.
NCOPY	2	Number of backup dumps to keep.
NDUMP	100_8	Number of dumps or directives. NDUMP must be less than 500_8.
NUMARF	1	Number of duplicate ARF copies.
TDEN	4	Tape density for dumps; any of the following. Value — Description 1 — 556 cpi 2 — 200 cpi 3 — 800 cpi 4 — 1600 cpi 5 — 6250 cpi This parameter is in deck COMKIPR.
TDTR	40_8+10_8*DTTP+TDEN	Tape format definition.
TTIGL	5000	Length of table that contains the transaction entries to be ignored during DMREC recovery.

Parameter	Default	Description
TLOGL	100	Number of files in database.
TVSNL	40	Number of VSNs allowed.
WBUFL	4001B	Length in words of buffer used to contain data read from a block on the after-image recovery file. Size depends on the installation parameters CRMARB and CMDM, and on the maximum record length specified for the database files in the xxJ file (refer to the TAF Reference Manual).

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 resident terminal capsule is Z721. At least one terminal capsule must be 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 build call to TERMLIB changes the default resident terminal capsule from only Z721 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

General installation parameters related to the common products are defined within the common deck IPARAMS, included in Product Texts.

The default values of the IPARAMS configuration parameters are defined with the CEQU or CMICRO macros so that you can insert all modifications at one place. The CEQU and CMICRO macros define symbols conditionally; that is, they are effective only if the variables have not been previously defined. Therefore, any modifications you make must precede them. Insert all changes to IPARAMS at IPARAMS.15.

Modifications to be applied to products TEXT and TEXTIO should be applied only in the procedure TEXT.

To obtain a listing of all installation parameters in IPARAMS as well as the micros set by the unique parameters, run a job similar to the following:

```
job command.
USER,INSTALL,INSTALL.
BEGIN,PRDIN,INSTALL,PRDNAME=TEXT,DISK=0.
UPDATE,Q.
COMPASS,A,I,S=0.
GTR,INSTALL,Z1.PROC/TEXT
COPYBF,Z1,OUTPUT.
--eor--
*COMPILE IPTEXT
--eoi--
```

The CSET, ECS, and MFT unique parameters also set these micros and eliminate the need to create a USER file for several of the micros.

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
C64	Selects the CDC graphic 64 character set.
A64	Selects the ASCII graphic 64 character set.
C63	Selects the CDC graphic 63 character set.
A63	Selects the ASCII graphic 63 character set.

This parameter sets the IP.CSET symbol; these products reference IP.CSET.

AAM 2	FCL 5
APL 2	FORTRAN 5
BASIC 3	Query Update 3
COBOL 5	Update 1
COMPASS 3	8-Bit Subroutines 1
FCL 1	
FCL 2	

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 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, 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 875, or 990. This parameter sets the MODEL, HF.LIST, and IP.CMU symbols; most common products reference the MODEL micro.

NOTE

Do not add user code to set the symbols IP.CSET, IP.MECS, IP.CMU, MODEL and HF.LIST since they are set by the above parameters. To specify your own values for HF.LIST, IP.CMU, and the MODEL micro, specify MFT=0 and apply user code to set these symbols.

INSTALLATION PARAMETERS

The following list constitutes the extent of installation-changeable symbols in IPARAMS.

Parameter	Default	Description
IP.CSET		Defines the character set to be used throughout the system. This parameter is controlled by the unique parameter CSET on the call to the TEXT installation procedure. The character set selected determines the collating sequence to be used.

To select the ASCII graphic 64-character set, specify a value of IP.C64.2 for the IP.CSET parameter.

To select the CDC graphic 63-character set, specify the following two parameter definitions:

```
IP.C63    EQU   IP.C64.1
IP.CSET   EQU   IP.C63
```

To select the ASCII graphic 63-character set, specify two parameter definitions:

```
IP.C63    EQU   IP.C64.2
IP.CSET   EQU   IP.C63
```

The following products reference IP.CSET:

AAM 2	FCL 5
APL 2	FORTRAN 5
BASIC 3	Query Update 3
COBOL 5	Update 1
COMPASS 3	8-Bit Subroutines 1
FCL 1	
FCL 2	

Parameter	Default	Description
IP.MECS	1	If you specify a nonzero value, extended memory is available for use when loading user programs. If you specify zero, user programs loaded by Loader cannot use extended memory. You must specify a nonzero value to allow user programs to use extended memory. Also, extended memory is available only if UEC is defined by the EQPDECK XM entry during deadstart. (Refer to the NOS 2 Analysis Handbook for information about EQPDECK entries.) This parameter is controlled by the unique parameter ECS on the call to the TEXT installation procedure.
OS.ID	NOS 2.5.1	System identification micro for displaying the operating system name and version number in generated program binaries. Most common products reference the OS.ID micro.

MFT PARAMETER DETAILS

The IP.CMU, MODEL, and HF.LIST micros are all set by the MFT parameter. To determine the relationship between MFT and these micros, consult a listing of the TEXT build procedure in DECKOPL.

Parameter	Default	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. A value for IP.CMU is set by the unique parameter MFT on the call for the TEXT installation procedure. COBOL 5 references IP.CMU.
MODEL		This parameter, which selects the mainframe model type, is controlled by the unique parameter MFT on the call to the TEXT installation procedure. All values of the model micro are supported. The default is for the CYBER 74. Here are the allowable values: 71, 72, 73, 74, 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 875, or 990. This parameter sets the MODEL symbol; most common products reference the MODEL micro.

Parameter	Default	Description
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 addition to the MODEL micro, since use of various hardware features by the products is conditional on HF.LIST. HF.LIST controls the following:

Entry	Description
C	Compare/move unit (CMU) hardware is present.
L	For model 176, LCME is present. For models 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 875, and 990, UEM is present only if defined during deadstart. Both LCME and UEM are kinds of memory for which direct access instructions (014 and 015) are defined.
Sn	Stack size; n specifies the size of the longest possible instruction stack program loop in words. If the mainframe being described has no stack, omit this entry. n can be either of the following.

n	Description
7	74 and 6600
10	175, 176, 740, 750, 760, 865, and 875

Parameter	Default	Description
		Entry / **Description**
		Px — Type of central processor; x can be one of the following values.

x	Description
S	6200, 6400, 6500, 71, 72, 73, 171, 172, 173, 174, 720, 730, 810, 815, 825, 830, 835, 840, 845, 850, 855, and 860; serial type CPU, etc.
74	6600, 6700, and 74
175	175, 740, 750, and 760
176	176
740	740
750	750
760	760
865	865
875	875
990	990

The processor type defaults to PS if HF.LIST is defined but the processor type is omitted.

Parameter	Default	Description
PSD		The central processor's exchange package contains a PSD register; model 176 only.
CRW		Central memory read/write operations are performed for 660/670 instructions; models 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 875, or 990.

Here are the default values selected for the HF.LIST parameter as set by the unique parameter MFT on the call to the TEXT installation procedure:

MODEL Micro Value	HF.LIST Default String
71	PS
72	C,PS
73	C,PS
74	P74,S7
171	PS
172	C,PS
173	C,PS
174	C,PS
175	P175,S10
176	P176,S10,L
720	C,PS
730	C,PS
740	P740,S10
750	P750,S10
760	P76,S10
810	C,PS,CRW,L
825	C,PS,CRW,L
830	C,PS,CRW,L
835	PS,CRW,L
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
875	P875,S10,CRW,L
990	P990,S64,CRW,L
Any other	PS

Duplicate parameter entries (such as two Px entries) are not allowed.

When defining HF.LIST for products intended to be run on more than one mainframe, you can use the central processor type PS, P74, or P175 and include stack size (even if some of the mainframes do not have a stack). You must not include C and L unless those features exist on all of the mainframes in the configuration. The resulting products do not necessarily perform optimally on any one of the mainframes, but they perform better on a parallel processor (such as a 175) if that processor type is set in HF.LIST.

UPDATE - COMMON MEMORY MANAGER VERSION 1, CYBER COMMON UTILITIES, AND UPDATE VERSION 1

The following Update features are available through assembly options. You can modify them by deleting the appropriate entry in the range UPDATE.703 through UPDATE.711. An attempt to use these features when the option is not assembled causes Update to issue error messages. For example, when PMODKEY is not set, the PULLMOD statement is not recognized as a legal directive.

Parameter	Default	Description
DECLKEY	Enabled	Enables DECLARE directive.
CHAR64	Enabled	Declares 64-character set Update program library output.
PMODKEY	Enabled	Enables PULLMOD statement and G option.
AUDITKEY	Enabled	Allows audit functions.
EDITKEY	Enabled	Allows merge and edit functions.
OLDPLKEY	Enabled	Enables Update to read both old-style and new-style old program libraries.
EXTOVLP	Enabled	Enables detection of four types of overlap involving two or more cards in a correction set.
DYNAMFL	Enabled	Declares dynamic table expansion. When this option is assembled, Update automatically expands tables as required and dynamically requests NOS to change the user field length to accommodate the additional table area. At the end of the run, the field length is reduced to that requested by the user.

Common Memory Manager (CMM) uses symbol definitions from common deck CMMCOM. The symbols defined in IPTEXT that specify the operating system are also used. You can change the following CMMCOM installation parameters for CMM.

Parameter	Default	Description
DEFVER	0	Defines which of two versions of CMM is to be used by default. Value — Parameter 0 — A version without error checking (FAST) is used. 1 — An error checking version (SAFE) is used.
FLF	2000B	When variable block code is not present (only fixed blocks exist), this value is used as a default by the field length reduction algorithm. The amount of free space above the highest fixed block is reduced to FLF central memory words.
FLINC	2000B	When field length is increased by CMM, this value is used as a default increase above the minimum amount needed.

CHAPTER 8

SYSGEN

SYSGEN 8

INTRODUCTION

This chapter describes the SYSGEN installation command in the following sections:

- Loading Permanent Files
- SYSGEN Functions
- SYSGEN Maintenance
- SYSGEN Validations

LOADING PERMANENT FILES

This section gives information on loading one or more permanent files from the permanent file tapes. If you have not deadstarted your new system, you will need to load the RECLAIM database as described under Loading the RECLAIM Database in this section.

CALLING SYSGEN

When loading permanent files, you may choose to have files installed directly to the appropriate user name or you may load files to a different user name for examination or modification and then move them to the correct user name at a later time. For example, you may load file NAMSTRT directly on user name NETOPS, or as an alternative, load it to user name INSTALL for modification, then move it to user name NETOPS later. 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 in this chapter for complete information.

To call SYSGEN from DIS, use this command format:

```
USER(user name,password)        -OR-      SUI(user index)
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 for 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 THE RECLAIM DATABASE

SYSGEN(INIT) is used to put the database on disk for full-product orders and SYSGEN(UPGRADE) is used for component-only orders. Both of these SYSGEN functions may be executed from either an interactive terminal or from DIS at the system console.

To install a full-product order:

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

2. To load files from an interactive terminal, log in to your installation user name, INSTALL, to set up the RECLAIM database. To load files from DIS at the system console, first do a USER or SUI command to your installation user name.

3. Enter the following commands from your terminal:

   ```
   LABEL,TAPE,VSN=vsn,D=density,LB=KU,F=I.
   COPYEI,TAPE,SYSTEM,V.
   UNLOAD,TAPE.
   BEGIN,SYSGEN,SYSTEM,INIT,UPGRADE.
   ```

   Replace vsn with the volume serial number of the tape; replace density with the tape density.

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 chapter. However, if you wish to use SYSGEN during multiple terminal sessions or from user names other than INSTALL, you must do the following before executing the SYSGEN command:

```
GET,SYSGEN/UN=INSTALL.
```

This gets the newly released version of SYSGEN, which in turn, gets the newly released PFGLIB from file PRODUCT. This is also on user name INSTALL. (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 only:

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 INSTALL 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

   You must do 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 if you do not have one) with information about the files on the permanent file tape. Once you have set up the RECLAIM database on your installation user name, you may proceed to load files from it.

LOADING THE FILES

- 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
NAMSTRT	Installs all products that update the NAMSTRT file.
LIBRARY	Installs all required files to user name LIBRARY.
SUBSYS	Installs all subsystem startup files (excludes NAMSTRT).
CDCNET	Installs components of CDCNET. When executed interactively, this group does not install all files; additional steps are required. Consult the CDCNET section of chapter 7.
OTHER	Miscellaneous products.
SITE	Installs standard installation site tools.

Consult the SYSGEN procedures for exact details on what files belong with what groups. Note that there is some overlap, for example, PFGFSE1 is used as part of SUBSYS, LIBRARY, and SITE.

SYSGEN(FILES)

Use this function to load all the permanent files from the permanent file tapes that are released with a component order. This function loads all the files from the permanent file tapes except for the RECLAIM database. Also, this function does not create a validation file. Use SYSGEN(FILES) after using the SYSGEN(UPGRADE) function when you are installing a component order.

Here is the format for calling SYSGEN(FILES):

```
X.SYSGEN(FILES,vsn)
```

Replace vsn with the VSN listed on the external tape label.

SYSGEN(FULL)

The SYSGEN(FULL) function loads all the permanent files for a standard installation. First, SYSGEN(FULL) loads your RECLAIM database and sets up validation files (if you have none). Next, SYSGEN(FULL) loads the permanent files. The types of files loaded include all subsystem startup procedure files, help files, online manuals, and other permanent files associated with specific products. If you are performing a customized installation, refer to SYSGEN(SOURCE) in this subsection. Here is the format for calling SYSGEN(FULL):

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

SYSGEN(RECLAIM)

This function loads files from the permanent file tapes and leaves them as local files. This function should not be called from DSD but can be called from DIS or an interactive terminal. Here is the format for the call:

```
SYSGEN(RECLAIM,f1,f2,f3,f4,f5)
```

Replace f1 through f5 with the names of the files you want to load from the dump tapes. (You can load up to five files with each call.)

NOTE

To use SYSGEN(RECLAIM), you must have previously loaded the RECLAIM database for the dump tapes to user name INSTALL. (This is done automatically by the SYSGEN(FULL) or SYSGEN(SOURCE) function.)

SYSGEN(SOURCE)

This function loads files from the permanent file database and enables you to begin a customized installation. This function can be run from DSD, DIS, or from an interactive terminal. Here is the format for calling the function:

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

Include the keyword CCP if you want to begin a customized installation of CROSS/CCP at the CCPVAR step (refer to chapter 7).

SYSGEN(SOURCE) sets up the RECLAIM database and the files DECKOPL, COMMOD, CODEPL, MISCPL, NAMSTRT, USERBPS, INSTALL, and DBUBIN. These are set up on the same user name as the SYSGEN(SOURCE) is executing from. SYSGEN(SOURCE) also puts a copy of RECLAIM on an indirect access permanent file.

Once SYSGEN(SOURCE) has completed, you can use RECLAIM to examine various release materials.

SYSGEN 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. Here is the format to call the function:

```
X.SYSGEN(COPYTAP,aaaaa,dl,numpf,dst,cip)
```

Parameter	Description
aaaaa	Specifies the first 5 characters of the VSN of the permanent file tape.
dl	Specifies the density of the release tapes.
numpf	Specifies the number of permanent file tapes to copy. Specify 0 if you do not want to copy any of the tapes.
dst	Specifies whether to copy the deadstart tape. Enter YES or NO.
cip	Specifies whether to copy the CIP tape. Enter YES or NO.

If you specify that permanent file tapes should be copied, the function writes a new label 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=dd,xx.
```

Replace vsn with the volume serial number of the first permanent file tape; replace dd with the tape density; replace xx with MT for a 7-track tape or NT for a 9-track tape. RECLAIM will read in all of the tapes and rebuild 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). Here is the format for the SYSGEN(DST) function:

```
SYSGEN(DST,old,lgo,new,input,dd)
```

Parameter	Description
old	Name of the base deadstart file. This deadstart file should be local to your job or you should supply the dd parameter so that SYSGEN can request a deadstart tape to use as the base. If a tape is requested, the VSN is ODT. This file can be a pre-assigned magnetic tape file.
lgo	Name of the file which contains the binaries to update file old. If lgo is not local, SYSGEN looks for a permanent file with that name. You can specify 0 (zero) if no binaries are to be updated.
new	Name of file to receive the new deadstart file. If no file is assigned and the dd parameter is specified, a tape request for VSN NDT will be made. This file can be a preassigned magnetic tape.
input	Name of a file containing directives for the LIBEDIT that SYSGEN performs. Specify 0 (zero) if you have no LIBEDIT directives.
dd	Tape density for any tapes to request. If you don't specify a tape density, no tape requests are made and SYSGEN only deals with local files.

SYSGEN(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 in this chapter. You must execute SYSGEN(MOVE) from the system console. Here is the format for calling SYSGEN(MOVE):

```
X.SYSGEN(MOVE,pfn,un,ct,ac,m)
```

Parameter	Description
pfn	Name of the permanent file to be moved. The file must reside on installation user name INSTALL.
un	User name to receive files. Valid user names and passwords are listed in table 8-1.
ct	Specifies the permanent file category. Enter PU for public, PR for private, and S for semiprivate. The default is PR.
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.
m	Assigns read permission to user name INSTALL. Enter PERMIT to assign read permission. The default is to not assign read permission. Use PERMIT only when ct is set to PR.

SYSGEN(SWAP)

This function creates a new deadstart tape with the DEBUG/TRACE versions of NAM, RBF, and PSU or a 63-character set version of NIP5870. Here is the format for calling this function:

```
X.SYSGEN(SWAP,density,p1,p2,p3)
```

Parameter	Description
density	Specifies the density of the new deadstart tape.
p1,p2,p3	Specify the products to be swapped. You can specify RBF, NAM, NIP, and PSU. Any combination can be used.

NOTE

Be sure to keep a copy of the deadstart tape that contains the original binaries.

SYSGEN(UPGRADE)

The SYSGEN(UPGRADE) function merges the binaries you receive for a component order with your existing system. You can use SYSGEN(UPGRADE) to perform any of these functions:

- Merge the component-order binaries with a tape copy of your system and then create a new deadstart tape.
- Merge the component-order binaries with a local file copy of your system and then create a new deadstart local file. (You can then write the file to tape or create a disk deadstart file.)
- Update the RECLAIM database without merging the binaries.

To merge binaries with a tape copy of your system and create a new deadstart tape, use this format:

```
SYSGEN(UPGRADE,OLD,NEW,vsn,d1,d2)
```

To merge binaries with a local file copy of your system and create a new local deadstart file, use this format:

```
SYSGEN(UPGRADE,old,new,vsn,d1)
```

To update the RECLAIM database without merging the binaries, use this format:

```
SYSGEN(UPGRADE,0,0,vsn,d1)
```

Parameter	Description
old	Name of local file containing the binaries of the current system.
new	Name of local file to receive the merged binaries.
vsn	Volume serial number of the CDC-supplied permanent file dump tape. The VSN is listed in the media number field on the external tape label.
d1	Tape density of the permanent file dump tape.
d2	Tape density of both the old and new deadstart tapes. The old deadstart tape is requested with the VSN of ODT; the new deadstart tape is requested with the VSN of NDT.

SYSGEN MAINTENANCE

The SYSGEN procedures are maintained in deck SYSGEN on DECKOPL. To create a new set of SYSGEN procedures, execute this command:

```
BEGIN,SYSGEN,INSTALL.
```

The SYSGEN procedures consist of a NOS procedure SYSGEN and a user library PFGLIB on the deadstart file. The BEGIN (SYSGEN,INSTALL) command adds SYSGEN to the PRODUCT and GLOBLIB files. (If GLOBLIB and PRODUCT are attached and a LIBRARY(GLOBLIB) has been done, all SYSGEN functions execute from PRODUCT and GLOBLIB.)

If you need to modify SYSGEN (rather than the running system), 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 SYSGEN modifica/tions.

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 methods. Before you can use these procedures, 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 will be created. Passwords for existing SYSGEN user names will be changed to match those in the table. Any user name not needed by SYSGEN will not be 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 may want to 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 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 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.

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	INSTALL	CCP, CDCNET, CROSS, DUAL STATE, FSE, MAP, NAM5, NOS
KB100DC	TAFPASS	TAF
LIBRARY	LIBRARY	DCL, FSE, NOS, XEDIT
MANUALS	MANUALS	NOS Online Manuals
NETADMN	NETADMN	CCP, CDCNET, NAM5
NETOPS	NETOPSX	CDCNET, ITF, NAM5, PSU, PTF/QTF, RBF5
NVE	NVEX	Dual State installed.
SSPOT	STATION	NSS
SUBFAM0	SUBFAM0	MSE
SYSTEMX	SYSTEMX	CDC2, DUAL STATE, FSE, IAF, MAP, MCS, MSE MSS, NAM5, NOS, NSS, RHF, TAF

CHAPTER 9

INSTALLATION COMMANDS, PARAMETERS, AND PROCEDURES

INSTALLATION COMMANDS, PARAMETERS, AND PROCEDURES 9

INTRODUCTION

The following commands, parameters, and procedures used during installation are detailed in this chapter:

- 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. Also refer to SYSGEN(SOURCE) in chapter 8 and COMMOD Procedure in this chapter.

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 parameters values for your site.
3. Replace file COMMOD.
4. Use the SETUP procedure to build a new INSTALL file from DECKOPL that contains your modifications to COMMOD. For example:

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

The following 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.

Value	Description
HI	556 characters per inch (cpi) (7-track).
HY	800 cpi (7-track).
HD	800 cpi (9-track).
PE	1600 cpi (9-track).
GE	6250 cpi (9-track).

NOTE

This parameter only affects tape requests for deadstart tape and density for output PLs when DISKINS=NO and OUTPRD=YES.

Parameter	Description
DISKINS (called from common deck COMDISK)	Controls the type of installation. The default is DISKINS=NO. **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.
IA (called from common deck COMIA)	Specifies how an installation job is submitted for execution. If you do not specify the keyword IA, the installation procedure submits a batch job for execution unless the installation procedure is already executing as a batch job. If the job origin type is not batch, you can specify the keyword IA to cause immediate execution of a build job. Specifying IA also causes the following: • If the job origin type is interactive, the file OUTPUT is assigned to mass storage and at job completion or abnormal termination is renamed IAOUT. If you want to assign the output back to your terminal rather than mass storage, enter this command: ASSIGN,TT,OUTPUT. • If the job is running from DIS, the installation procedure will run from DIS and will not submit a batch job. • The build procedure issues a LIBRARY,GLOBAL command when the job is finished executing. This causes the local file GLOBAL (if one exists) to be made the global library. This feature allows you to have your own global library named GLOBAL in effect at a terminal to run build jobs, and to have GLOBAL remain in effect after the build jobs are completed.

Parameter	Description
ICHG (called from common deck COMIUN)	Specifies the valid charge and project number values for the installation user name. Here is a sample value that causes the command CHARGE(1187,594N321) to be executed: *N=$1187,594N321$. This parameter is significant only if IUCHG=YES.
IFAMILY (called from common deck COMIUN)	Specifies the default value for the alternative family name. Set this value if you are not installing on the system default family. This parameter is significant only if IUCHG=YES.
IPW (called from common deck COMIUN)	Specifies the password for IUN (installation user name). The default password is INSTALL. The password is used by SUBPROC. This parameter is significant only if IUCHG=YES.
IUCHG (called from common deck COMIUN)	Controls the source of USER and CHARGE commands for batch job submittal. The default value is YES. Value / Description YES — The parameter values specified by IUN, ICHG, IPW, RESOUR, and PCKNAM are used to generate a USERCHG file. NO — You must create a USERCHG file (containing appropriate USER, CHARGE, RESOURC, and PACKNAM commands) for the build jobs. The file can be local, direct, or indirect.
IUN (called from common deck COMIUN)	Specifies the installation user name. The default installation user name is INSTALL. This parameter is significant only if IUCHG=YES.
LIST=filename (called from common deck COMLIST)	Specifies the file for assembly or compilation listing output. If filename is OUTPUT, the listing is printed with the installation listing. If you specify any other filename, use the TOLIST parameter to specify the destination for the output file. Also, the file cannot be a magnetic tape file if the installation procedure uses the FORTRAN compiler.
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 jobs. The default is NO.

Value	Description
YES	The build jobs 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 volume serial number [VSN]).
NO	Output program libraries are not generated or written.

NOTE

If you do not apply user code to change program library common decks for those products also used as auxiliary PLs and you do not want to write output PLs, you can use the following parameter settings:

```
OUTPRD=NO
PSRIN=nnn
PSROUT=nnn
```

nnn defaults to the current release level. The input PLs are then used as auxiliary PLs and no product output files are written. Binaries are still stored in file PRODUCT. The released DECKOPL has the parameters set in this manner.

PCKNAM (called from common deck COMIUN)	Specifies the pack name and type if all files used in the installation process are to reside on an auxiliary disk pack. The entry format is (*N=$PACKNAM,pname/R=typr.$). The default is *. This parameter is significant only if IUCHG=YES.
PFGPN (called from common deck COMPFG)	Specifies the auxiliary pack name where PFGprdname files are written. The default option is to write these files to the catalog under which the installation job is executing. (Also refer to the PFGPR and USEPFG parameters.)
PFGPR (called from common deck COMPFG)	Specifies the auxiliary pack type where PFGprdname files are written when the files are to reside on an auxiliary pack. (Also refer to the PFGPN and USEPFG parameters.)

Parameter	Description
PRID (called from common deck COMPRID)	Specifies the printer identification for routing print files from verification jobs.
PSRIN (called from common deck COMIN)	Specifies the 3-digit number identifying the PSR level of this release. The default is the current release level.
PSROUT (called from common deck COMOUT)	Specifies the 3-digit number identifying the PSR level which is appended to the requested file name for the output PL files. The default is the current release level. If you do not change this value and OUTPRD=YES, the output PLs will overwrite the input PLs.
RESOUR (called from common deck COMIUN)	Specifies the format of the RESOURC command to use; for example, RESOURC(DJ=1,GE=1). This parameter is significant only if IUCHG=YES.
TOBLD=option (called from common deck COMTOB)	Specifies the build listing disposition and determines whether the job OUTPUT goes to the wait queue. The default is TOBLD=PRINT.

Value	Description
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.
PRINT	Job output is printed at the central site.

Parameter	Description
TOLIST=option (called from common deck COMTOL)	Specifies the assembly list file routing and whether file LIST goes to the wait queue. The default is TOLIST=NONE; if the job fails, the list file is discarded.

Value	Description
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.
NONE	The list file, if defined, is local to the job and is discarded when the job terminates.

NOTE

When you route listings to the wait queue, these files are counted in the total number of jobs validated for your user name. Also, if you want to specify different options for the LIST, TOBLD, and TOLIST functions, you can recode the procedures named JOBPASS and JOBFAIL in DECKOPL.

Parameter	Description
UN1 (called from common deck COMDISK)	Specifies the alternative user name for input source files. Set UN1 to a value only if you are rebuilding under a user name other than IUN. The default value is 0; it specifies that the user name under which the job is executing is used.

NOTE

If you specify a user name for UN1, DISKINS must be set to NO if the source program libraries have not been preloaded.

UN2 (called from common deck COMDISK)	Specifies the alternative user name for output source files. The default is to store the files using the same user name as the job is executing under.

NOTE

If you specify a user name with the UN2 parameter, OUTPRD must be set to NO.

USEPFG (called from common deck COMPFG)	Specifies whether you want to save the PFGprdname files created by SYSGEN 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.

Value	Description
YES	All PFGprdname files are saved regardless of whether they existed previously.
NO	Only PFGprdname files that previously existed are replaced with current PFGprdname files. If a file does not exist, no new permanent file is created.

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,param,param,...,param.
```

NOTE

Include all the keyword=value parameters before using the keyword parameters.

Parameter	Description
REP=n	Specifies the number of additional copies to be printed. The default is 0.
MOD=fn	Specifies the name of a modification set file to be added to the PL for listing purposes. The file can be a local file or a direct or indirect access permanent file. The PL is not permanently updated.
EXPAND=nn	Specifies whether common deck calls will be expanded. You can specify YES or NO. The default is YES.
MODIF	Includes the default Modify list options on the listing.
UMODE	Lists only modified decks; used in conjunction with the MOD=fn parameter.
NODATA	Suppresses expansion of data sections for verification jobs.

Here are some examples of calls to the DECKLIS procedure:

```
BEGIN,DECKLIS,INSTALL,MOD=COMMOD,UMODE.

BEGIN,DECKLIS,INSTALL,NODATA.
```

DISK INSTALLATIONS

If you want to perform your installation with the source program libraries on disk allowing the installation procedures to manage disk files for you, change the DISKINS parameter in file COMMOD to YES.

You can preload all the source files to disk before beginning this process, or you can let the installation procedures load them one at a time as they are needed. If you want to preload your files, refer to Using PFLOAD, later in this chapter.

Each build job calls the procedure PRDIN. At the first reference to the product release file, PRDIN looks for a local file of the correct name (for example, NAM5). If the file is not local, PRDIN attempts to attach the file from disk. When the file is not on disk, PRDIN issues a RECLAIM command to load the file from the permanent file tapes and then store the file on disk.

The disk files are given the name of the product source file concatenated with the value of PSRIN. For example, the disk file for NAM5 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 must specify the DI parameter to force PFLOAD to load files to the proper user index.
- You must specify the FM and DD parameters if you are loading a family device.
- You must specify the PN parameter if you are loading an auxiliary device.
- If your site is running an archival system, you should specify the OP=Z and UD parameters; Control Data maintains these files using Mass Storage Extended (MSE).
- 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 2 Analysis Handbook.

All source program libraries are loaded with names of the format xxxxpsrin where xxxx is the 4-character product name (such as TEXT and 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 PFG 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 contain four parameters. Use the first two parameters to define the disk pack name and type for storage of the input source file. Use the other two parameters to define the assignment for the output source file which you can assign to a different disk pack.

Each product source file that also serves as an auxiliary program library requires a second common deck defining its disk pack location. The second common deck is used in build jobs that use the auxiliary program library. This second deck describes the auxiliary program library disk location. It is named using the following format:

COMXa

Parameter	Description
COMx	Identifies an auxiliary common deck in DECKOPL.
a	One-character, identical to the last character of this common deck used by PR DIN 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
PNO=PACKX      PN3=PACKX
PRO=DJ         PR3=DJ
```

Table 9-1. Source File and Auxiliary PL Assignments

Common Deck	Product Job	Auxiliary Common Deck†	Source File	Required Aux PLs††	Common Deck	Product Job	Auxiliary Common Deck†	Source File	Required Aux PLs††
COMD1	AAM2	COMX1	AAM2		COMD3	AP1I		AP1I	X
COMD1	APL2		APL2	X	COMD3	AP1L		MMCL	
COMD1	BAM		BAM1	1	COMD3	CCL		CCL1	X,1
COMD1	BASIC3		BAS3	X,1	COMD3	CEDIAG		CEDG	X
COMD1	BINEDIT		BINE	1	COMD3	DUAL		DUAL	X,1
COMD1	BIT8		BIT8		COMD3	FORMAT		FMAT	X
COMD1	CCG	COMX1	CCG1		COMD3	HCD		ZHCD	
COMD1	CDCS2		CDCS	X,1	COMD3	ITF		ITF1	X,3
COMD1	CID		CID1	X	COMD3	MMCL		MMCL	
COMD1	COBOL5		COB5		COMD3	MSSI		MSSI	X
COMD1	COMPASS	COMX1	CPS1		COMD3	MTS		ZMTS	X
COMD1	DCL		DCL2		COMD3	NSS		NSS1	X
COMD1	DDL	COMX1	DDL3	1	COMD3	RHC	COMX3	RHC1	
COMD1	FCL1		FCL4	1	COMD3	RHF		RHF1	X,3
COMD1	FCL2		FCL4	1	COMD3	RHP		RHP1	X,3
COMD1	FCL5		FCL5	1	COMD3	TDU		TDU1	X
	FCS3		FCS3						
COMD1	FDBF		FDBF	1		FSE			X
COMD1	FORM		FORM			HS10			X
COMD1	FTN		FTN4	1		IAF			X
COMD1	FTNT5		FTI4	1		MMF			X
COMD1	FTN5		FTN5	1		MSE			X
COMD1	F4S		F451	1		MSS			X
						NIP5870			X
COMD1	LCS3		LCS3			NOS			X
COMD1	LOADER		LDR1			NOS2B			X
COMD1	PASCAL		PASC	X		OSLIB			X
COMD1	PMD		PMD5	1		OSTEXT			X
COMD1	QU3		QU31	X,1		RDFEX			X
						TAF			X
COMD1	SORT5		SRT5			TOOLS			X
COMD1	SYMPL		SYMP			TRACER			X
COMD1	TEXT	COMX1	TEXT			XEDIT			X
COMD1	TEXT10		TX10						
COMD1	UPDATE		UPD1	1					
COMD2	CHA1		CHA1	X,2					
COMD2	MCS		MCS1	X,2					
COMD2	NAM5	COMX2	NAM5	X					
COMD2	RBF5		RBF5	X,2					

†A common deck name in this column identifies the source file as one also used as an auxiliary PL.
††Each letter or digit in this column refers to common decks for auxiliary PLs needed by this build job. For example, X means COMXOPL, 1 means COMX1, 2 means COMX2.

GENDST PROCEDURE

Use the GENDST procedure to merge the binaries on file PRODUCT with a base deadstart file and generate a new deadstart file. The GENDST procedure ensures that there are no conflicts between IAF and RDF on the new deadstart file.

Here is the format of the GENDST call:

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

Parameter	Description
SYSTEM=odt	Local file name for the old deadstart file. If file odt is preassigned, it becomes the base deadstart file; otherwise, a tape label request is issued with a VSN=ODT.
NEW=ndt	Local file name for the new deadstart file. If file ndt is preassigned, the new deadstart file is written on it; otherwise, a tape label request is issued with a VSN=NDT. The default file name is NEW.
D=density	Tape density option. If this parameter is omitted, the value in deck COMDEN set in COMMOD is used. The default is PE. The option applies to both odt and ndt. Here are the values you can specify for tape density:

Value	Description
HI	556 cpi (7-track).
HY	800 cpi (7-track).
HD	800 cpi (9-track).
PE	1600 cpi (9-track).
GE	6250 cpi (9-track).

Parameter	Description
LIST=list	Local file name for the listing file. The default file name is OUTPUT. If you run GENDST interactively and want to save the listing file or do not want it to appear at the terminal, specify a different file name.

Run a job similar to the following to add site-provided binaries and deadstart decks to the new deadstart file. Create file USERD so it contains the LIBEDIT directives (refer to NOS 2 Reference Set, Volume 3) to make these additions.

Job	Comments
job command.	
USER,username,password,familyname.	
GET,USERD.	USERD contains the LIBEDIT directives.
GET,lfn=pfn.	lfn (permanent file name is pfn) contains the modified deadstart decks. lfn must appear in the USERD file as *FILE lfn.
BEGIN,GENDST,INSTALL.	Parameters are not required if you use the system defaults.

MISCGET PROCEDURE

All code that may correct a specific user site problem but that has not been fully tested is contained on the file MISCPL. This file, if released, was created during the setup of installation files. The modifications are properly formatted (Update or Modify) for the intended program library.

To list the modification set headers and the decks containing calls to common decks containing modification sets, use the following commands:

```
BEGIN,MISCGET,INSTALL,HISTORY.
ROUTE,USER,DC=PR.
```

You can extract code from MISCPL using any of the following methods:

- To extract all modification sets for a product, use the following command:

  ```
  BEGIN,MISCGET,INSTALL,PRD=name.
  ```

  Replace name with the name for the deck containing a call for each modification set available for the product.

- To extract one modification set, use the following command:

  ```
  BEGIN,MISCGET,INSTALL,MOD=modset.
  ```

  Replace modset with the name of the required modification set.

- To extract selected modification sets, use the following commands:

  NOTE,lfn,NR.+.modname$_1$+.modname$_2$...+.modname,
  BEGIN,MISCGET,INSTALL,MISCIN=lfn.

These calls to MISCGET append the modification sets to the local file USER. You should then save file USER as a permanent file for later use by the corresponding build procedure.

REPORT PROCEDURE

To obtain statistics on all completed installation jobs, run a job that contains this command. The job output indicates the resources used for each installation job and whether the job passed or failed.

```
BEGIN,REPORT,INSTALL,XC.
```

NOTE

The REPORT procedure uses the direct access file REP 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 build procedures, and creates DIRFILE with entries for those ULIBs. The build procedures subsequently add binaries to the file PRODUCT and directives to file DIRFILE. Only those user libraries used by subsequent build jobs are required to remain on file PRODUCT after GENDST creates a new deadstart tape.

To use the RESETP procedure to reduce the size of files PRODUCT and DIRFILE, follow these steps:

1. Run GENDST to create a new deadstart tape.

2. After writing the new deadstart tape, enter this command:

   ```
   BEGIN,RESETP,INSTALL.
   ```

SETUP PROCEDURE

The SETUP procedure generates the permanent file INSTALL, which contains all installation procedures. The SYSGEN(FULL) function (as well as other SYSGEN functions) initially loads files from the permanent file tapes and creates INSTALL. SETUP can also perform the following functions:

- Create INSTALL from DECKOPL with optional modifications against DECKOPL.

- Rename file INSTALL to a specified name.

- Create a 63-character set version of procedures on INSTALL.

- Convert DECKOPL and INSTALL to a 63-character set format.
- Replace DECKOPL with an updated DECKOPL.

Here is the format for calling SETUP:

BEGIN,SETUP,INSTALL,params.

Parameter	Description
NEWPL	Replaces DECKOPL with modified DECKOPL. If NEWPL is omitted, DECKOPL is not replaced.
MOD=filename	Applies modsets from specified file to DECKOPL; the file can be a local file or a permanent file. If you change any default parameters in COMMOD and you also have your own local changes to DECKOPL, append your changes to file COMMOD. Then use MOD=COMMOD in your procedure call.
DF63	Selects 63-character set version of installation procedures for inclusion on file INSTALL.
CV63	Converts DECKOPL to 63-character set format.
INSTALL=filename	Creates or replaces procedure file specified. The default is INSTALL. If you omit the INSTALL keyword, procedure file INSTALL is not created or replaced.

Example:

The following sample call to the SETUP procedure applies modifications from file COMMOD against the DECKOPL installation procedures and creates a new INSTALL procedure file. DECKOPL is not updated.

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

APPENDIX A

GLOSSARY

GLOSSARY A

Account Dayfile

The account dayfile provides a history of system usage. It also provides information necessary for accurate billing and system usage and analysis.

APRDECK

The APRDECK (Auxiliary Mass Storage Parameter Deck) is a text record on the deadstart file. The entries in this deck specify flawed disk areas. A flawed area is one that cannot be read from or written to by the system.

ASCII

American National Standard Code for Information Interchange. The standard character set and code used for information interchange between computer systems.

Auxiliary Device

Mass storage device that is not part of a permanent file family. Auxiliary devices can contain direct or indirect access permanent files.

Batch Critical Update

A binary correction to CDCNET or dual state.

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

The number of the data channel on which a peripheral device controller can be accessed.

Character

Unless otherwise specified, references to characters in this handbook are to 7-bit ASCII code characters.

Checkpoint

The process of writing to a magnetic tape or mass storage file a copy of your job's central memory, the system information used for job control, and the names and contents of all assigned files that are identified in a CHECKPT request.

CMRDECK

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

Coldstart

Procedure used to deadstart if the tape or disk controller has not yet been loaded with controlware or the controlware is not running.

Command

A sequence of words and characters that call a system routine to perform a job step. A command is sometimes called a control statement.

Communication 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 data base.

Data Channel

One of the 9 to 24 channels (12-bit) by which information passes between the peripheral processors and peripheral devices. Refer to Channel Number.

Dayfile

A chronological file created during job execution which forms a permanent accounting and job history record. Dayfile messages are generated by operator action or when 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.

Deadstart

The process of initializing the system by loading the operating system library programs and any of the product set from magnetic tape or disk. Deadstart recovery is reinitialization after system failure.

Deadstart Sequencing

The execution of a selected set of commands before normal system job scheduling is enabled.

Default Value

A fixed value supplied by the system for a missing parameter.

Detached Job

An interactive service class job removed from control of the interactive subsystem. It may or may not continue to execute, depending on the presence of commands in the command buffer or an active job step. Control is regained by recovering the EJT entry for the job.

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 that you can access 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 (Job Display)

A system peripheral processor program similar to system display (DSD) program. DIS provides communication between a job in central memory and the operator at the console, and permits the operator to control execution of the program through the console keyboard.

Disk

A unit composed of one or more flat, circular plates with magnetic material on both sides that is used to store large amounts of data or programs.

Disk Pack

A group of disks with magnetically encoded information.

Display Code

A 6-bit character code set used to represent alphanumeric and special characters.

Displays

Two console screens or a split screen used to display system and job information, operator messages, and contents of central memory. Through the console keyboard, the operator can control the operation of the system. The displays are identified by alphabetic characters; some used frequently are the job status (B), system files (H), and dayfile messages (A).

Disposition Code (DC)

A 2-character mnemonic indicating the destination queue and format for processing a file named on a ROUTE function.

DSD (Dynamic System Display)

The operating system program that provides communication between the operator and the system by accepting control information typed on the console keyboard and by displaying information pertinent to all jobs known to the system. DSD is permanently assigned to peripheral processor 1.

ECS

Extended Core Storage. Refer to Extended Memory.

EQPDECK

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

Equipment Number

A number from 0 to 7 which identifies the setting on a peripheral device controller.

Equipment Status Table (EST)

A central memory resident table listing all defined equipment, parameters affecting their operation, and the status of the equipment.

EST

Refer to Equipment Status Table.

EST Ordinal

The number designating the position of an entry within the equipment status table (EST) established at each installation. Devices are identified in operator commands by EST ordinals.

Extended Core Storage (ECS)

Optional additional memory. ECS contains 60-bit words; it has a large amount of storage and fast transfer rates. ECS can be used only for program and data storage, not for program execution. Special hardware instructions exist for transferring data between central memory and ECS.

Extended Memory

An extension to central memory which is physically located outside of the machine. Also referred to as extended core storage (ECS), extended semiconductor memory (ESM), large central memory extended (LCME), or unified extended memory (UEM).

Family Device

Mass storage permanent file device associated with a specific system. A family may consist of 1 to 63 logical devices. Normally, a system runs with one family of permanent file devices available. However, additional families may be introduced during normal operation. This enables users associated with the additional families to access their permanent files via the alternate family.

Family Name

A designation that the installation may give to a group of permanent file devices.

Family Ordinal

An index into the family ordinal table (FOT). The family ordinal is used to identify a unique family.

Family Ordinal Table (FOT)

A central memory resident table used to associate family names with family ordinals.

Field Length (FL)

The area in central memory allocated to a particular job. The only part of central memory that a job can directly access.

File

1. A set of information that begins at beginning-of-information (BOI), ends with end-of-information (EOI), and can be referenced by a local file name.

2. That portion of a multiple file terminated by an end-of-file (EOF).

3. Data recorded on a magnetic tape beginning after HDR1 label and ending before an EOF1 label.

FOT

Refer to Family Ordinal Table.

IAF

Refer to Interactive Facility.

Indirect Access File

A NOS permanent file that you access by making a temporary copy of the file (GET or OLD commands). You create or alter it by saving or substituting the contents of an existing temporary file (REPLACE or SAVE commands). Compare with Direct Access File.

Interactive Facility (IAF)

An application that provides a terminal operator with interactive processing capability. The interactive facility makes terminal input/output and file input/output appear the same to an executing program.

Interactive Origin Job

A job initiated from an interactive (time-sharing) terminal.

Job Sequence Name (JSN)

The unique system-defined name assigned to every executing job or queued file. The JSN is a string of 4 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, job status indicates why it was rolled out.

JSN

Refer to Job Sequence Name.

LCF

Refer to Local Configuration File.

LCN

Refer to Loosely Coupled Network.

LID

Refer to Logical Identifier.

Local Configuration File (LCF)

A file in the host computer system containing information on the logical makeup of the communication elements of the host. The file contains a list of the application programs available for execution in the host computer, and the users that can access it. This is a NOS direct access permanent file.

Local NPU

An NPU that is connected to the host via a coupler. A local NPU always contains a host interface program (HIP) for processing block protocol transfers across host/local NPU interface.

Logical Identifier (LID)

A 3-character alphanumeric string used to identify a particular mainframe. LIDs are identified by your site.

Login

The procedure used to gain access to the system.

Logout

The procedure used to end a terminal session.

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.

NDI

Refer to Network Device Interface.

NDL

Refer to Network Definition Language.

Network

An interconnected set of network elements consisting of a host and one or more NPUs and terminals.

Network Access Device (NAD)

The primary element in a loosely coupled network. Each NAD connects a computer system to the network.

Network Access Method (NAM)

A software product that provides a generalized method of using a communications network for switching, buffering, queuing, and transmitting data. NAM is a set of interface routines used by a terminal servicing facility for shared access to a network of terminals and other applications, so that the facility program does not need to support the physical structures and protocols of a private communications network.

Network Configuration File (NCF)

A network definition file in the host computer. This file contains information on network elements and permissible linkages between them. The status of the elements described in this file is modified by the NPU operator in the course of managing the network. This is a NOS direct access permanent file.

Network Definition Language (NDL)

The compiler-level language used to define the network configuration file and local configuration file contents.

Network Device Interface (NDI)

The standard CDCNET DI variant that transfers data between networks (for example, between two local area networks, between a local area network and a communications line, or between a local area network and a public data network).

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 network access method (NAM) application program, which 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, which 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 which must appear in a specific order.

Order-Independent

Used to describe items which need not appear in any specific order. Parameters, particularly those with keywords, may be order-independent.

Origin Type

A job attribute that indicates how a job entered the system. The four origin types are interactive origin, batch origin, remote batch origin, and system origin.

OUTPUT

The system-defined file which, by default, contains all the ouput from job processing. It is also known as the print or punch file.

Peripheral Microcode

Special type of software that resides in a peripheral controller. The controlware defines the functional characteristics of the controller.

Peripheral Processor (PP)

The hardware unit within the host computer that performs physical input and output through the computer's data channels.

Peripheral Processor Unit (PPU)

First-level peripheral processor (FLPP). A PPU is contained in the mainframe in a multimainframe environment and operates synchronously with the mainframe.

Permanent File

A mass storage file that is catalogued by the system so that its location and identification are always known to the system. They are protected by the system from unauthorized access according to privacy controls specified when they are created.

Physical Identifier (PID)

The unique 3-character identifier of a specific host.

PID

Refer to Physical Identifier.

PP

Refer to Peripheral Processor.

PPU

Refer to Peripheral Processor Unit.

Printer Support Utility (PSU)

Operating system software used to control the 533/536 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.

Queue Priority

An attribute associated with input and output files. If all other factors are equal, queue priority is used to select the best file for processing.

Queued File

An input, print, plot, or punch file that has an entry in the queued file table (QFT). It is not assigned an EJT entry and is waiting to be selected for processing.

QFT

Refer to Queued File Table.

Queued File Table (QFT)

A central memory resident table containing a 4-word entry for all active input and output queue files.

RBF

Refer to Remote Batch Facility.

Remote Batch Facility (RBF)

A network application that provides a terminal operator with remote batch processing capabilities. RBF transfers input and output files between remote batch devices and NOS.

Remote Batch Job

A job submitted from a remote batch terminal.

Remote Host Facility (RHF)

A central processor program that executes at a system control point. It performs data buffering and switching and is the intermediary between application programs and the network.

Remote NPU

A network processing unit linked to a host computer through other network processing units.

RHF

Refer to Remote Host Facility.

Rollout

The removal of jobs from central memory to mass storage before execution is complete, so the control point and central memory can be assigned to another job. A job is rolled out when it is waiting for an external event, when its control point is needed by a higher priority job, or when it exceeds its central memory time slice.

Rollout File

A file containing a job (and system information) that has been temporarily removed from the main processing area of the system.

Scheduling Priority

An attribute associated with an executing job available for job scheduling. Scheduling priority is used to select the best executing service class job for processing.

SC

Refer to Service Class.

Service Class (SC)

An attribute associated with a queued file or executing job. Service class determines how the system services the job.

Suspended Job

An interactive job placed in an inactive state. Processing is stopped immediately and recovery information is copied to the rollout file. Processing is resumed as if no interruption took place, if the job's EJT entry is recovered.

SYSLIB

Refer to System Library.

System Job

A job brought to a control point by the operator.

System Library (SYSLIB)

The collection of tables and object language programs residing in central memory or on mass storage which are necessary for running the operating system and its product set.

System Origin Job

A job entered at the system console.

TCU

Refer to Trunk Control Unit.

Trunk

The communication line connecting two network processing units.

Trunk Control Unit (TCU)

The hardware part of a network access device (NAD) that interfaces with a network trunk.

UJN

Refer to User Job Name.

Unit Number

The setting of a hardware device. Used when more than one hardware unit can be connected to a controller.

User Job Name (UJN)

A 1- to 7-character alphanumeric name you specify to replace the system-defined job sequence name (JSN) for a queued file or executing job.

Volume Serial Number (VSN)

A 1- to 6-character identifier that identifies the volume of magnetic tape to the system.

VSN

Refer to Volume Serial Number.

Warmstart

Procedure used to deadstart if the tape or disk controller is loaded and the controlware is running.

APPENDIX B

FILE FORMATS

FILE FORMATS B

ORDER OF PRODUCTS

The order in which products appear on the deadstart tape is now controlled by DECKOPL in each build deck. The basic order is alphabetical.

NOTE

All ULIBs are in library 4.

Arrangement of deadstart tape libraries:

Library	Description	Contributing Build Decks
1	Fixed order (CTI)	
2	Fixed order	NOS, HSIO, MMF, DISK MICROCODE
3	NOS	NOS, NOS2B, NIP5870
4	All ULIBs	Many
5	Miscellaneous items	TDU, PFTF, Misc. MICROCODE
6	AAM2	AAM2
7	(empty)	
8	APL2	APL2
9	BAM1	BAM
10	BAS3	BASIC3
11	BINE	BINEDIT
12	BIT8	BIT8
13	CCL1	CCL
14	CDCS	CDCS2
15	CEDG	CEDIAG
16	CHA1	CHA1
17	CID1	CID
18	CML1	CML
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	MMCL/MSSI	MMCL/MSSI/AP1I
34	MCS1	MCS
35	(empty)	

Library	Description	Contributing Build Decks
36	MMF1	MMF
37	MSE1	MSE
38	MSS1	MSS
39	NAM5	NAM5/NAM5D
40	(empty)	
41	NSS1	NSS
42	OSTX	OSTEXT
43	PASC	PASCAL
44	(empty)	
45	PMD5	PMD
46	PSU1	
47	QU31	QU3
48	RBF5	RBF5/RBF5D
49	RHF1	RHF
50	(empty)	
51	RHP1	RHP
52	(empty)	
53	SRT5	SORT5
54	SYMP	SYMPL
55	TAF1	TAF
56	TEXT	TEXT
57	TXIO	TEXTIO
58	TOOL	TOOLS
59	TRCE	TRACER
60	UPD1	UPDATE
61	XEDT	XEDIT
62	(empty)	
63	(empty)	

The following is a list of all files on the permanent file tape(s). All source files are listed first, followed by all PFG files (files processed by SYSGEN). At the end of the list are files which are received only on a component order.

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 tape(s), follow these steps:

1. Load the RECLAIM database as described under Loading Permanent Files in chaper 8.

2. Enter this command from your installation user name:

   RECLAIM,Z./LIST,UN=NS2psrin.

   Replace psrin with the release level.

This will generate 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 will not be requested.

SOURCE FILES

All source files end with the value psrin, the current NOS release level. For example, the AAM2 source file 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.

Source Files:

File Name	Associated DECKOPL Procedure Name	Format Notes
AAM2psrin	AAM2	
APL2psrin	APL2	File 1. Program library.
		File 2. APLLIB. Relocatable of APL.
		File 3. APLPROD. Absolute of APL.
		File 4. TAPLTST. APL verification jobs.
		File 5. TAPLOUT. Sample output for file 4.
		File 6. NEWSF. APLNEWS, news file.
		File 7. FILESYS. Workspace, file functions.
		File 8. FILES2. Workspace, file functions.
		File 9. APLNEWS. Workspace, information.
		File 10. CATALOG. Workspace, information.
		File 11. WSFNS. Workspace, general functions.
		File 12. TAPLWS. Workspace for APL verification.
		File 13. Reserved.
AP1Ipsrin	AP1I	AP1 online diagnostics for MAP III and MAP IV.
		File 1. FTN5 version of source PL.
		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	

File Name	Associated DECKOPL Procedure Name	Format Notes
CCPBpsrin	CCPPH1, CCPBLB	File 1. Program library for CCP base. File 2. Expand and autolink binaries. File 3. Mux firmware (phase 1) load module. File 4. Dump bootstrap module. File 5. System autostart (SAM) load module. File 6. Symbol table for dump bootstrap load module. File 7. Combined binary library (BCMB) for generation of phase 2 variant load modules. File 8. Compiler listing for expand and autolink. File 9. CCP listing for phase 1 load module. File 10. CCP listing for dump bootstrap load module. File 11. CCP listing for system autostart load module. File 12. CCP assembly listing for BCMB. File 13. CCP Pascal listing for BCMB.
CCPDpsrin	CCPPH1	CCP online diagnostics - sequential PL.
CCPRpsrin	CCPPH1	CCP Remote Concentrator (LIP) - sequential PL.
CCPTpsrin	CCPPH1	CCP 3270 Bisync TIP - sequential PL.
CDCSpsrin	CDCS2	
CEDGpsrin	CEDIAG	
CHA1psrin	CHA1	
CID1psrin	CID	
COB5psrin	COBOL5/COBOL5a	File 1. Sequential PL. Files 2,3,4. Used by the COBOL5Q build procedure.
CPS1psrin	COMPASS	
CRSSpsrin	CROSS	File 1. Source program library. File 2. Absolute binary for Cross. File 3. Empty. File 4. Empty. File 5. Pascal cross-reference. File 6. MPLINK. File 7. MPEDIT. File 8. CCP Pascal compiler. File 9. Pascal compiler bootstrap. File 10. Empty. File 11. Cross program listing.
DCL2psrin	DCL	File 1. Program library. File 2. SEGLOAD directives. File 3. Absolute for DCUPD. File 4. Absolute for DCSEL. File 5. Absolute for DCRPT. File 6. Absolute for DCRET. File 7. Absolute for DCCONVT. File 8. Absolute for DCUTL. File 9. Absolute for DCIDX. File 10. Absolute for DCGEN. File 11. Absolute for DCCONGN.
DDL3psrin	DDL3	
FCL4psrin	FCL1/FCL2	
FCL5psrin	FCL5	

File Name	Associated DECKOPL Procedure Name	Format Notes
FCS3psrin	FCS3	File 1. Program library. File 2. Binary data for FORTRAN file conversion processor (FCP) verification. File 3. Binary data for COBOL FCP verification. File 4. Absolute load module CBLFCP1 for COBOL FCP. File 5. Absolute load module CBLFCP2 for COBOL FCP. File 6. Absolute load module FTNFCP1 for FORTRAN FCP. File 7. Absolute load module FTNFCP2 for FORTRAN FCP. File 8. FORTRAN binary syntax file CBLFCPM for COBOL FCP. File 9. FORTRAN binary syntax file FTNFCPM for FORTRAN FCP.
FDBFpsrin	FDBF	
FMATpsrin	FORMAT	
FORMpsrin	FORM	
FTI4psrin	FTNTS	
FTN4psrin	FTN4	
FTN5psrin	FTN5	
F451psrin	F45	
ITF1psrin	ITF	
LCS3psrin	LCS3	File 1. Program library. File 2. Absolute load module for LCS. File 3. FORTRAN binary syntax for LCS. File 4. Absolute load module COUP for COSY-to-Update file conversion. File 5. Absolute load module COPYCOB for COBOL-COPY-library file conversion. File 6. Binary data file for COSY-to-Update file conversion (file 1 of 2). File 7. Binary data file for COSY-to-Update file conversion (file 2 of 2).
LDR1psrin	LOADER	
MCS1psrin	MCS	
MMCLpsrin	MMCL/AP1L	
MSSIpsrin	MSSI	
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, NOS2B, FSE, HSIO, IAF, MMF, MSS, TAF, TOOLS, TRACER, MSE, and XEDIT. OPLpsrin also contains the input PLs for the build procedures RDFEX, NOS2B, NIP5870, TDU, TERMLIB, OSLIB, and OSTEXT.
PASCpsrin	PASCAL	
PFTFpsrin	PFTF	File 1. Program library.
PMD5psrin	PMD	
QU31psrin	QU3	
RBF5psrin	RBF5/RBF5D	
RHC1psrin	RHC	

File Name	Associated DECKOPL Procedure Name	Format Notes
RHF1psrin	RHF	
RHP1psrin	RHP	
SRT5psrin	SORT5	
SYMPpsrin	SYMPL	
TDU1psrin	TDU	File 1. TDUTOOL 1. File 2. TDUTOOL 2. File 3. TDUTOOL 3.
TEXTpsrin	TEXT	
TXIOpsrin	TEXTIO	
UPD1psrin	UPDATE	
VSM1psrin	CCPVAR	CCP VARIANT VSM1 File 1. CCP variant (phase 2) load module. File 2. CCP PICB load module. File 3. Symbol table for CCP variant and PICB. File 4. CCP listing for CCP variant build (MPLINK). File 5. Listing of CCP variant build (MPEDIT). File 6. Listing of PICB build (Pascal, MPLINK, and MPEDIT).
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.
VSM1psrin	CCPVAR	CCP variant SM1; same format as file VSM1.
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.
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	SYSGEN Function/Notes
PF 6AAAA		LOADUSE	SYSGEN data file
PFGAPL2	APL2	APL2	File 1. Empty file. File 2. APLLIB. Relocatable of APL. File 3. APLPROD. Absolute of APL. File 4. TAPLTST. APL verfication job. File 5. TAPLOUT. Sample output for file 4. File 6. NEWSF. APLNEWS, news file. File 7. FILESYS. Workspace, file functions. File 8. FILES2. Workspace, file functions. File 9. APLNEWS. Workspace, information. File 10. CATALOG. Workspace, information. File 11. WSFNS. Workspace, general functions. File 12. TAPLWS. Workspace for APL verfication.
PFGAP1I	AP1IAP1I		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	SYSGEN Function/Notes
PFGAP1L	AP1L	AP1L	MMCL files for AP1 Online Diagnostics. File 1. AP1LIB1; Channel Coupled MAP IV-2X. File 2. AP1LIB2; Channel Coupled MAP IV-2X. File 3. AP1LIB3; Channel Coupled MAP IV-2X. File 4. AP1LIB1; ECS/CMI Coupled MAP III/IV. File 5. AP1LIB2; ECS/CMI Coupled MAP III/IV. File 6. AP1LIB3; ECS/CMI Coupled MAP III/IV.
PFGBCU1			File 1. Installation procedures. File 2. RECLAIM database. File 3. RECLAIM dump of CDCNET permanent files.
PFGBCU2		BCU2	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.
PFGCDCS	CDCS2	CDCS	CDCS startup procedure.
PFGCHA1	CHA1	CHA1	File 1. Installation procedure. File 2. CHA1 message template file, HTF_id_psrout. File 3. CHA1 response file, HRF_id_psrout. File 4. CHA1 log file, HLF_id_psrout. File 5. CHA1 NAMSTRT records.
PFGCHA2		CHA2	File 1. Installation procedure. File 2. NETOU template file, TF_id_vvvv.
*PFGCODE		CODE	CODEPL.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	SYSGEN Function/Notes
*PFGCRSS	CROSS	CRSS	Cross Binary Install Permanent Files. File 1. Empty. File 2. Absolute binary for CROSS. File 3. Empty. File 4. Empty. File 5. Pascal cross-reference. File 7. MPEDIT. File 8. CCP Pascal compiler. File 9. PASCAL compiler bootstrap.
PFGCML1		CML1	Refer to CML Reference Manual for specific file format information.
*PFGCSTD		CSTD	File 1. COMPASS coding standards. File 2. SYMPL coding standards.
*PFGDBU1		DBU1	For use by QU3. (DBUBIN)
PFGDCL2	DCL	DCL2	Permanent files for Data Catalogue 2 V2.0. File 1. Absolute for DCUPD. File 2. Absolute for DCSEL. File 3. Absolute for DCRPT. File 4. Absolute for DCRET. File 5. Absolute for DCCONVT. File 6. Absolute for DCUTL. File 7. Absolute for DCIDX. File 8. Absolute for DCGEN. File 9. Absolute for DCCONGN.
PFGDCNS		DCNS	File 1. Installation procedures. File 2. RECLAIM database. File 3. RECLAIM dump of CDCNET permanent files.
*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 file. File 2, record 2. FSEHELP file. File 3, record 3. FSTEACH file. File 4, record 4. FSEPROC file. File 5, record 5. SMFSTAT file.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	SYSGEN Function/Notes
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. MSE startup procedure. File 2, record 2. MSE slave startup procedure. File 3, record 3. MSE sample configuration file.
PFGMSSI	MSSI	MSSI	MSSI 3 Permanent Files. File 1. MAPCMI startup procedure. File 2. MAPECS startup procedure. File 3. MAPCH startup procedure. File 4. MSSIP - misc. MSSI/MMCL procedures. File 5. MJOBSPL - MSSI test jobs PL.
PFGMSS1	MSS	MSS1	MSS startup procedure.
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.

File Name	DECKOPL Procedure Name	SYSGEN Procedure Name	SYSGEN Function/Notes
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	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).
*PFGPRPT		PRPT	History idents of NOS PSR code in current release.
PFGPSUD		SWAP	Printer Support Utility binaries. SYSGEN procedure SWAP is used to load these binaries. File 1. 64-character set trace binaries.
PFGPSU1		PSU1	File 1. PSU NAMSTRT startup record. File 2. PSU default settings file, EVFULFN. File 3. Contains PSU relocatables.
PFGSRB1		SRB1	File 1. SRB in ASCII 6/12 format. File 2. (Optional) Feature notes bulletin in ASCII 6/12 format. File 3. (Optional) User IMPACT statements in ASCII 6/12 format.
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.
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
PRODUCT	Contains all binaries for the deadstart tape.
DSTDIR	Contains LIBEDIT directives for the products in file PRODUCT.
BLDLIB	Contains a procedure to properly update all ULIBs from binaries in file PRODUCT.

APPENDIX C

63-CHARACTER SET INSTALLATION

63-CHARACTER SET INSTALLATION C

To install a system with the 63-character set format, perform the customized installation described in this appendix.

INSTALLATION PROCEDURE

To install a 63-character set system, follow these steps:

1. Load the files necessary for installation (refer to chapter 6).

2. If you plan to run verification procedures, refer to appendix F for information on merging the verification job file with DECKOPL.

3. Execute the following command to convert the Modify-formatted DECKOPL procedures to the 63-character set format:

   BEGIN,SETUP,INSTALL,CV63,NEWPL.

4. 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.

5. Run the UCOMMOD procedure to recreate file COMMOD. Modify parameters, if necessary.

   BEGIN,UCOMMOD,INSTALL.

   Use the new COMMOD file to set up DECKOPL installation defaults.

   BEGIN,SETUP,INSTALL,DF63,INSTALL,MOD=COMMOD.

6. Run the SEED procedure. Refer to chapter 6 for more information.

7. Create file USER for the MODOPL procedure, if needed.

8. Execute the MODOPL procedure. The composite OPL is created in the 63-character set format. Refer to chapter 6 for more information.

9. 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.

10. Include the parameter CSET=C63 or CSET=A63 on the call to build TEXT.

11. 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 user name INSTALL, are installed by SYSGEN and must be converted to 63-character set format because they are not recreated from the build jobs:

- USERBPS
- NDLDATA
- PSRRPT
- ASCII 6/12 SRB

PRINTER SUPPORT UTILITY

Relocatable binaries of the Printer Support Utility (PSU) are contained on the released permanent file tapes. You can use these relocatables plus the DECKOPL procedure PSULINK to generate 63-character set versions of PSU. Refer to the PSU section in chapter 7 for complete information about the PSULINK procedure.

5870 INSTALLATION

On a 63-character set system, the build procedure NIP5870 must be rerun and 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,dn,NIP)
```

Replace dn 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.

APPENDIX D

THE REMOTE DIAGNOSTIC FACILITY

THE REMOTE DIAGNOSTIC FACILITY D

The Remote Diagnostic Facility (RDF) allows you to use an interactive terminal when your networks are not yet installed. Thus, you can use the RDF as an interface for using a NOS editor to create your deadstart deck files or configuration files.

To initiate RDF, follow these steps:

1. When you are deadstarting the system, make an EQPDECK entry that defines the two-port multiplexer to which the terminal you want to use is connected. (Check with your customer engineer (CE) to ensure that the terminal's baud rate agrees with that set on the two-port multiplexer. Refer to the NOS 2 Analysis Handbook for the syntax of the entry.)

2. After you have deadstarted the system, check the operator E,A. display for the equipment status of the two-port multiplexer. If the status is OFF, enter this command:

   ```
   ON,nn
   ```

nn	The EST ordinal of the two-port multiplexer.

3. Enable the RDF subsystem by entering these commands:

   ```
   SUBSYST.
   L.ENABLE,RDF.
   L.END.
   ```

4. Now initiate RDF by entering this command:

   ```
   RDF.
   ```

5. Press the RETURN key on the terminal a few times; the login banner is displayed at the terminal.

Using RDF is similar to using IAF/NAM, with the following exceptions:

- To correct input, press the backspace key or CTRL-H.
- To delete input, press the ESCAPE key.
- To interrupt execution, press I or the BREAK key.
- To terminate execution, press S or enter STOP.
- To exit TEXT mode, press CTRL-C or the BREAK key.
- The typing-ahead feature is not available on RDF.
- Full Screen Editor (FSE) does not function properly.

APPENDIX E

SYSTEM CONFIGURATIONS

SYSTEM CONFIGURATIONS E

To minimize installation time and confusion and to establish conventions that will enable better CDC service/support to users, the preferred configurations have been defined in table E-1.

Complete configurations are not defined in table E-1 due to the differing customer requirements. Conventions are specified for a base system. Systems configured according to the conventions will provide the following benefits:

- Simplify the selection of peripheral equipment channels by the customer and customer engineering.
- Allow the customer to easily deadstart NOS using the pre-defined configurations released on the deadstart tape.

Table E-1. Preferred Configuration

Channel	CYBER 170/180-800 Models 815, 825, 835, 840, 845 850, 855, 860, 865, 875, 990	CYBER 180 Models 810, 830		
		a	b	c
0	+	RMS-ISD$_1$	RMS-ISD$_1$	RMS-ISD$_1$
1	ROTATING MASS STORAGE (RMS)$_1$	+	RMS$_1$	RMS-ISD$_3$
2	RMS$_2$	−	−	−
3	RMS$_3$	+	RMS$_2$	RMS-ISD$_4$
4	RMS$_4$	+	RMS$_3$	+
5	COMMUNICATIONS$_1$	−	−	−
6	COMMUNICATIONS$_2$	MT-ISMT$_1$	MT-ISMT$_1$	MT-ISMT$_1$
7	COMMUNICATIONS$_3$	COMMUNICATIONS	COMMUNICATIONS$_1$	COMMUNICATIONS
10	CONSOLE	CONSOLE*	CONSOLE*	CONSOLE*
11	MAG TAPE$_1$	+	+	+
12	UNIT RECORD	+	UNIT RECORD	+
13	+	−	−	−
20	+	RMS-ISD$_2$	RMS-ISD$_2$	RMS-ISD$_2$
21	RMS$_5$	+	RMS$_4$	+
22	RMS$_6$	−	−	−
23	RMS$_7$	+	+	+
24	RMS$_8$	+	+	+
25	+	−		−
26	+	ISD/ISMT	ISD/ISMT	ISD/ISMT
27	+	+	+	+
30	+	−	−	−
31	MAG TAPE$_3$	+	MAG TAPE$_1$	+
32	MAG TAPE$_4$	+	MAG TAPE$_2$	+
33	MAG TAPE$_2$	−	−	−

a: Specifies channel assignments for a 810/830 using only Intelligent Small Disk (ISD) (834s & 836s) and Intelligent Small Tape (ISMT) (639s) disk and tape peripherals.

b: Specifies channel assignments for a 810/830 with both ISD/ISMT peripherals and other CDC disk and tape peripherals.

c: Specifies channel assignments for the C180-810 and 830 with the 18102-2 channel option.

+: Indicates that the channel is available for CYBER channel peripheral connection.

−: Indicates that the channel is not available for CYBER channel peripheral connection.

*: 18002-1 console only.

1: This subscript indicates that you should use this channel for RMS/MAG tape/communications first.

2..8: These subscripts indicate additional channels that you may use.

APPENDIX F

INSTALLATION VERIFICATIONS

INSTALLATION VERIFICATIONS F

This appendix describes how to update DECKOPL with the verification procedure file and how to execute the verification procedures. This appendix contains the following sections:

- Merging Verification Procedure File with DECKOPL
- Product Verification

MERGING VERIFICATION PROCEDURE FILE WITH DECKOPL

The verification procedures exist in OPL format on file VJOBS. This file is loaded to user name INSTALL by SYSGEN(SOURCE). If you plan to use these procedures, they must be merged with DECKOPL. To do so, use the following commands:

```
CHANGE,DECKOLD=DECKOPL.
ATTACH,DECKOLD.
GET,VJOBS.
DEFINE,DECKOPL.
LIBEDIT,P=DECKOLD,B=VJOBS,N=DECKOPL,I=0.
RETURN,DECKOLD,VJOBS,DECKOPL.
```

If your cite installs a 63-character set version of the system, refer to appendix C. If your site installs a 64-character set version of the system, execute the following command:

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

PRODUCT VERIFICATION

The verification procedures cannot be executed until you have completed the installation and have deadstarted the system using the new deadstart tape. Execute the REPORT procedure to get statistics on all completed installation procedures (refer to chapter 9).

Note that not all products have verification procedures and that some products require special steps. Products requiring special steps are listed in order at the end of this appendix. (This includes verifications for CDCS2, MCS, NAM and CCP, PSU, RBF, and RHF.)

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

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

Parameter	Description
pname	pname is the verification procedure name of the product to be verified (refer to table F-1).
PRID=id	id is 0 to 60. This parameter specifies the printer for verification job output.

The following command calls all verification procedures on file INSTALL (refer to table F-1) except VCDCS2A, VCDCS2B, VMCS1A, VMCS1B, and VRHF.

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

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

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

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

Since VJOB submits all verification procedures except VCDCS2A, VCDCS2B, VMCS1A, VMCS1B, and VRHF, procedures that verify unordered products will fail.

Table F-1. Verification Procedures

Verification Procedure Name	Required Files	Comments	Verification Procedure Name	Required Files	Comments
VAAM2			VJOBS		Submits all verification jobs except VCDCS2A, VCDCS2B, VMCS1A, VMCS1B, and VRHF.
VAPL2			VLCS3	LCS3psrout	
VBAM †			VMCS1A		
VBASIC3			VMCS1B		MCS must be enabled to execute VMCS1B.
VBIT8			VMSSI		
VCDCS2A			VPASCAL		
VCDCS2B			VQU3		VQU3 must be run as a batch job.
VCID			VRHF		
VCOBOL5			VSORT5		
VCROSS	CRSSpsrout		VSYMPL ††		
VDDL3					
VFCS3	FCS3psrout				
VFDBF					
VFORM					
VFTN					
VFTNTS					
VFTN5					
VF45					

†Expect the following nonfatal error messages:
RM NOTE 1000 ON LFN TAPE 1
RM NOTE 1000 ON LFN TAPE 2

††Expect the following nonfatal error message:
1 MACHDEP ERRORS (SUPPRESSED)

CDCS2 VERIFICATION

To verify the installation of CDCS2, do the following:

1. Execute the following command.

   ```
   BEGIN,VCDCS2A,INSTALL,S=INSTALL.
   ```

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

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

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

   n is the control point.

3. Execute the following command.

   ```
   BEGIN,VCDCS2B,INSTALL.
   ```

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

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

   ```
   K,CDC.
   ```

5. Enter the following command to terminate CDCS2.

   ```
   K.TERM.
   ```

MCS VERIFICATION

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

1. Execute the following command:

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

   This job creates an application definition library file (ADLLIB) on the installation user name INSTALL.

2. To start MCS processing, enter:

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

   n is the control point number.

3. To compile a sample COBOL job and execute MCS-related verbs, execute the following command:

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

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

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

   n is the control point number.

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

```
VERIFY AOP
```

   as the MCS application name and

```
TERMINAL1
```

   as the symbolic name.

6. A verification message appears at the terminal. After verification is complete, enter:

```
CFO,MCS.IDLE
```

   to terminate MCS.

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

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

NAM AND CCP VERIFICATION

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

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

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

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

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

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

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

```
NAM VER ver - lvl
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

NOTE

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

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

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

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

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

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

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

```
LINE TEST BEGINS
..
```

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

```
TVF TEST COMPLETE
..
```

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

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

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

```
LOGGED OUT.
```

PSU VERIFICATION

Use the following steps to verify correct installation of PSU:

1. Initiate NAM as described under NAM5 - Network Access Method.

2. Enter the following commands at the system console:

   ```
   X.DIS.
   USER,INSTALL,PASSWORD.
   CATLIST.
   DROP.
   ```

3. Ready a printer connected to PSU.

4. The output from the DIS job should print at the printer connected to PSU. This printout consists of two banners, the CATLIST output, and the dayfile.

RBF VERIFICATION

Use the following steps to verify correct installation of RBF.

1. Prepare the following card deck. If necessary, supply the CHARGE command.

   ```
   job command.
   USER,username,password,familyname.
   NOTE.+1TOP OF PAGE+0ABCDEFGHIJKLMNOPQRSTUVWXYZ
   6/7/8/9
   ```

   A multipunch in column 1 (or for HASP terminals, /*EOI in columns 1 through 5).

2. Initiate NAM as described under NAM5 - Network Access Method. (It is assumed that RBF is defined as an application in COMTNAP.)

3. Log in from the console of any network-supported terminal with at least one card reader and one line printer, specifying RBF as the application and any user name permitted to use RBF (refer to the Remote Batch Facility Reference Manual). Follow all commands issued in this procedure by the message transmission key for the terminal class you are using (ETX or SEND for many batch terminal consoles.) RBF responds with a header line giving the version of RBF and the date and time of login, followed by READY on the following line.

4. Enter DIS. RBF displays the status of the batch devices associated with the terminal.

5. Place the card deck prepared in step 1 into the input hopper of the card reader and initiate reading (described in the Remote Batch Facility Reference Manual).

6. Enter the following command at the terminal console.

   ```
   GO,CRn
   ```

   n is the device number of card reader (from 1 through 7).

7. RBF reads the job in the card reader and responds with READY.

8. Enter DIS,PR at the terminal console. Because the system processes the job prepared in step 1 in little time, RBF should indicate that the job is in the print queue. If not, repeat this step until such an indication is received.

9. Ready the line printer, and enter the following command at the terminal console.

   GO,LPn

   n is the device number of line printer (from 1 through 7).

10. RBF prints the job's output on the line printer. The output consists of a banner page, a page with TOP OF PAGE printed on the first line of the page and the alphabet on the third line, and a page containing the job dayfile.

11. Enter BYE to exit RBF and log off of the system. The system issues a message indicating the time that has elapsed since connection to RBF.

    RBF ENDED yy/mm/dd. hh.mm.ss

    RBF CONNECT TIME hh.mm.ss

RHF Installation Verification

The VRHF procedure on DECKOPL can be used to verify the correct installation of Remote Host Facility (RHF) and permanent and queue file transfer applications. Before executing the procedure, ensure that the logical identifier table (LID) table is set up correctly. For detailed information about LIDs, the LID table, and physical identifiers (PIDs), refer to the NOS 2 Analysis Handbook.

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

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

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

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

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

INDEX

COMMENT SHEET

MANUAL TITLE: CONTROL DATA® NOS Version 2 Installation Handbook

PUBLICATION NO.: 60459320 **REVISION:** H

NAME: ______________________________

COMPANY: ______________________________

STREET ADDRESS: ______________________________

CITY: ____________ **STATE:** ____________ **ZIP CODE:** ____________

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

☐ Please Reply ☐ No Reply Necessary

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND TAPE

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA

Technology and Publications Division
ARH219
4201 North Lexington Avenue
Saint Paul, MN 55126-6198

FOLD FOLD

CUT ALONG LINE