60494300

CONTROL DATA

NOS/BE VERSION 1 INSTALLATION HANDBOOK

CDC® COMPUTER SYSTEMS:
CYBER 180
CYBER 170
CYBER 70
MODELS 71, 72, 73, 74
6000

REVISION RECORD

REVISION	DESCRIPTION
A (11-01-75)	Original release.
B (03-05-76)	Revised to reflect PSR Summary level 420. CYBER Loader 1.1, COMPASS 3.3, CYBER Record Manager 1.4, FORTRAN Extended 4.6, Sort/Merge 4.4, and BASIC 3.1 are updated. COBOL 5 is a new product with this release.
C (07-16-76)	Revised to reflect PSR Summary level 430. New features include 844 Factory Format, 844-41/44 Support, 844 Expander Support, Job Management and System Control Point enhancements, INTERCOM Restart, and Enhanced Station performance improvement. CCP Support Software 1 is replaced by CYBER Cross System 1.1. The operating system level is now NOS/BE 1.1; CYBER Loader is now version 1.2.
D (12-03-76)	Revised to reflect PSR Summary level 439. A new feature, Common Memory Manager, is included. Products updated are SYMPL 1.2, COBOL 4.6, Sort/Merge 4.5, CDCS 1.1, QU 3.1, DDL 2.1, DBU 1.1, and COBOL 5.1.
E (04-25-77)	Corrects various technical and typographical errors, adds information on CYBER Control Language, 67x tape units, 844 full-track recording mode, programmable format control for 580 printers, and system idle mode; and documents NOS/BE 1.2 at PSR level 447/446. Page numbering in this manual is changed as follows: Material formerly in the introduction is now in part I, part I material is now part II, and part II material is now in part III. This manual obsoletes all previous editions.
F (08-01-77)	Corrects various technical and typographical errors and documents NOS/BE 1.2 at PSR level 454.
G (12-09-77)	Corrects various technical and typographical errors and documents NOS/BE 1.2 at PSR level 461. Procedures to deadstart using a 7152 Mass Storage/Magnetic Tape Controller are included.
H (06-13-78)	Corrects various technical and typographical errors and documents NOS/BE 1.3 at PSR level 473/470. New products included with this release are CYBER Interactive Debug, PL/I, INTERCOM 5, and Communication Control INTERCOM. Products updated are UPDATE 1.3, COMPASS 3.5, CYBER Loader 1.4, CMM 1.1, BAM 1.5, AAM 2.0, FCL 4.7, FORTRAN Extended 4.7, COBOL 4.7, COBOL 5.2, Sort/Merge 4.6, CDCS 1.2, CDCS 2.0, DBU 1.2, DDL 2.2, DDL 3.0, QU 3.2, SYMPL 1.3, FORM 1.1, 8-Bit Subroutines 1.1, BASIC 3.2, ALGOL 4.2, and CCP 1.1. This revision also includes information for installing NOS/BE 1.3 on a model 176. This manual obsoletes publication number 60454830. Because of extensive changes to this manual, revision bars and dots are not used and all pages reflect the latest revision level. This edition obsoletes all previous editions.
J (10-20-78)	Revised to reflect PSR level 481. Features documented include the COMPASS common common decks and the new CCI installation procedure.
K (02-19-79)	Revised to reflect PSR summary level 488. Features documented include ALGOL 5, FORTRAN Data Base Facility, user direct access ESC swapping, and FNT threshold support.
L (07-20-79)	Revised to support NOS/BE 1.3 at PSR level 499. Features documented include Gemini load leveling, FORTRAN 5, EXPORT High Speed, Data Catalogue 2, and the addition of direct access and actual key to AAM Extended.
M (12-21-79)	Revised to reflect NOS/BE 1.4 at PSR level 508. Features documented include the 885 disk subsystem, the fast deadstart dump analyzer, common testing and initialization (CTI), and RMS deadstart. This revision also removes deadstart information (it is contained in the operator's guide) and corrects various technical and typographical errors.
Publication No. 60494300	

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
N (05-19-80)	Revised to reflect NOS/BE 1.4 at PSR level 518. Documents user capability to log error information for ECS errors. Corrects technical and typographical errors.
P (11-17-80)	Revised to reflect NOS/BE 1.4 at PSR level 530. Documents fixed assignment of user/terminal IDs and relocation of INTERCOM system tables from central memory resident to the managed buffer area. Corrects technical and typographical errors.
R (04-20-81)	Revised to reflect NOS/BE 1.5 at PSR level 538. Documents BCC tape structure, loading CTI and HIVS modules to disk, on-line binary patch, off-line binary patch, changes in the diagnostic routines, and CCI build process. Corrects technical and typographical errors.
T (11-23-81)	Revised to reflect NOS/BE 1.5 at PSR level 552. Documents CTI/MSL disk area utility, and Sort/Merge Version 5. Corrects technical and typographical errors.
U (05-17-82)	Revised to reflect NOS/BE 1.5 at PSR level 564. Revision U supports models 825, 835, and 855 and Remote Maintenance Facility (RMF) and Remote Host Facility (RHF) features. Various technical and editorial corrections are made. This edition obsoletes all previous editions.
V (11-22-82)	Revised to reflect NOS/BE at PSR level 577. Revision V documents support of models 865 and 875, and reflects the feature name change from Remote Maintenance Facility (RMF) to Remote Diagnostic Facility (RDF). It also documents changes to CTI and CML installation. Various technical and editorial changes are made.
W (02-20-84)	Revised to reflect NOS/BE at PSR level 604. There have been numerous deletions concerning information which is now to be found in the CYBER Initialization Package (CIP) User's Handbook.
Y (02-25-85)	Revised to reflect NOS/BE at PSR level 627. Various technical and editorial changes are made.
AA (02-21-86)	Revised to reflect NOS/BE at PSR level 650. Removes documentation of Sort/Merge 4, PL/I1 and ALGOL 605; STIMULATOR operating instructions, QTF initiation procedure parameters and CYBER utilities. Replaces documentation of CTI binary release with CIP install tape. Adds SPY, PRNTSPY, CIA and CPMET use and QTF configuration requirements. Various technical and editorial changes are made.
AB (12-15-86)	Revised to include NOS/VE Dual-State Preparation.

Publication No. 60494300

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	AB
3/4	AB
5	AB
6	AB
7	AA
8	AA
9	AA
10	AA
11	AA
12	AA
13	AA
14	AA
15	AA
16	AA
17	AB
18	AB
19	AB
Divider	-
I-1-1	AA
I-1-2	AA
I-1-3	AA
I-1-4	AA
I-1-5	AA
I-1-6	Y
I-1-7	AA
I-1-8	U
I-1-9	AB
I-1-10	AA
I-1-11	AA
I-1-12	AA
I-1-13	U
I-1-14	U
I-1-15	V
I-1-16	Y
I-1-17	AA
I-1-18	AA
I-1-19	V
I-1-20	V
I-1-21	AA
I-1-22	AA
I-1-23	AA
I-1-24	AA
I-1-25	AA
I-1-26	AA
I-1-27	AA
Divider	-
II-1-1	AA
II-1-2	AA
II-1-3	AB
II-1-4	AA
II-1-5	AA
II-1-6	U
II-1-7	W
II-1-8	AA
II-1-9	AA
II-1-10	AA
II-1-11	U

PAGE	REV
II-1-12	AA
II-1-13	U
II-1-14	AB
II-1-15	AA
II-1-16	Y
II-1-17	AB
II-1-18	AA
II-1-19	AB
II-1-20	U
II-1-21	U
II-1-22	U
II-1-23	AA
II-1-24	U
II-1-25	U
II-1-26	AA
II-1-27	AA
II-1-28	AA
II-1-29	AA
II-1-30	W
II-1-31	U
II-1-32	U
II-1-33	U
II-1-34	AA
II-1-35	AA
II-1-36	U
II-1-37	AA
II-1-38	U
II-1-39	AA
II-1-40	AA
II-1-41	U
II-1-42	U
II-1-43	U
II-1-44	W
II-1-45	U
II-1-46	W
II-1-47	V
II-1-48	U
II-1-49	W
II-1-50	U
II-1-51	AA
II-1-52	U
II-1-53	AA
II-1-54	AA
II-1-55	AA
II-1-56	AA
II-1-57	AA
II-1-58	AA
II-1-59	AA
II-1-60	AA
II-1-61	AA
II-1-62	AA
II-1-63	Y
II-1-64	W
II-1-65	W
II-1-66	W
II-1-67	W
II-1-68	W
II-1-69	W
II-1-70	W

PAGE	REV
II-1-71	W
II-1-72	W
II-1-73	W
II-1-74	W
II-1-75	W
II-1-76	W
II-1-77	Y
II-1-78	W
II-1-79	W
II-1-80	AA
II-1-81	AA
II-1-82	W
II-1-83	W
II-1-84	W
II-1-85	AA
II-1-86	AA
II-1-87	AA
II-1-88	AA
II-1-89	AA
II-1-90	AA
II-1-91/ II-1-92	AA
II-1-93	W
II-1-94	W
II-1-95	W
II-1-96	W
II-1-97	W
II-1-98	W
II-1-99	W
II-1-100	W
II-1-101	W
II-1-102	W
II-1-103	AA
II-1-104	W
II-1-105	W
II-1-106	AA
II-2-1	Y
II-2-2	U
II-3-1	U
II-3-2	U
II-3-3	U
II-4-1	AA
II-4-2	AA
II-4-3	Y
II-4-4	Y
II-4-5	Y
II-5-1	Y
II-5-2	AA
II-6-1	U
II-7-1	Y
II-7-2	V
II-8-1	U
II-8-2	AA
II-8-3	AA
II-8-4	AA
II-8-5	AA
II-8-6	AA
II-8-7	AA
II-8-8	AA

PAGE	REV
II-8-9	AA
II-8-10	AA
II-8-11	AA
II-8-12	AA
II-8-13	AA
II-8-14	AA
II-9-1	AA
II-9-2	W
II-9-3	U
II-9-4	U
II-9-5	U
II-10-1	Y
II-11-1	AA
II-12-1	Y
II-13-1	Y
II-14-1	AA
II-15-1	U
II-15-2	U
II-15-3	U
II-15-4	U
II-16-1	AA
II-16-2	U
II-16-3	AA
II-17-1	U
II-17-2	AA
II-17-3	U
II-17-4	U
II-17-5	U
II-17-6	U
II-17-7	U
II-18-1	Y
II-19-1	AA
II-19-2	AA
II-20-1	Y
II-21-1	AA
II-21-2	AA
II-22-1	U
II-23-1	Y
II-24-1	U
II-24-2	AA
II-24-3	U
II-24-4	U
II-24-5	U
II-25-1	U
II-25-2	U
II-26-1	U
II-26-2	AA
II-26-3	AA
II-27-1	Y
II-27-2	Y
II-27-3/ II-27-4	Y
II-27-5	Y
II-27-6	Y
II-27-7	U
II-28-1	AA
II-28-2	U
II-28-3	U
II-28-4	U

PAGE	REV
II-28-5	U
II-28-6	U
II-28-7	Y
II-28-8	Y
II-28-9	Y
II-28-10	AA
II-28-11	AA
II-28-12	AA
II-28-13	AA
II-28-14	AA
II-28-15	AA
II-28-16	U
II-28-17	U
II-28-18	U
II-28-19	U
II-28-20	U
II-28-21	V
II-28-22	U
II-28-23	U
II-28-24	W
II-28-25	W
II-28-26	AB
II-28-27	U
II-28-28	U
II-28-29	W
II-28-30	U
II-28-31	AB
II-28-32	AB
II-28-33	AB
II-28-34	AB
II-28-34.1/ II-28-34.2	AB
II-28-35	U
II-29-1	Y
II-29-2	U
II-29-3	U
II-29-4	U
II-29-5	U
II-29-6	U
II-29-7	U
II-29-8	U
II-29-9	U
II-29-10	U
II-30-1	AA
II-30-2	U
II-30-3	U
II-30-4	U
II-31-1	U
II-32-1	U
II-32-2	U
II-33-1	AA
II-34-1	U
II-34-2	U
II-35-1	U
II-36-1	AA
II-37-1	U
II-38-1	AA
II-38-2	U
II-39-1	U

PAGE	REV
II-39-2	U
II-40-1	U
II-40-2	U
II-41-1	AA
II-42-1	Y
II-42-2	AA
II-42-3	AA
II-42-4	AA
II-42-5	AB
II-42-6	Y
II-42-7	Y
II-42-8	AB
II-42-9	AB
II-42-10	AB
II-42-11	AA
II-42-12	Y
II-42-13	Y
II-42-14	AA
II-42-15	AB
II-42-16	AA
II-42-17	Y
II-42-18	AB
II-42-19	Y
II-42-20	Y
II-42-21	Y
II-42-22	Y
II-42-23	AA
II-42-24	AA
II-43-1	V
II-43-2	U
II-43-3	W
II-44-1	AB
II-44-2	AB
II-44-3	AB
II-44-4	AB
II-44-5	AB
II-44-6	AB
II-44-7	AB
Divider	-
III-1-1	V
III-1-2	U
III-1-3	U
III-1-4	U
III-1-5	U
III-1-6	U
III-1-7	U
III-1-8	V
III-1-9	V
III-1-10	U
III-1-11	U
III-1-12	U
III-1-13	U
III-1-14	Y
III-1-15	AA
III-1-16	AA
III-1-17	AA
III-1-18	AA
III-1-19	AA
III-1-20	AA
III-1-21/ III-1-22	Y
III-1-23	Y
III-1-24	U
III-1-25	U
III-1-26	AA
III-1-27	AA
III-1-28	U
A-1	AA
A-2	W
A-3	U
A-4	Y
A-5	Y
A-6	Y
A-7	AA
A-8	AA
A-9	AA
Index-1	AB
Index-2	AB
Index-3	AB
Index-4	AB
Index-5	AB
Index-6	AB
Index-7	AA
Comment Sheet	AB

PREFACE

This handbook describes installation information for the CONTROL DATA® Network Operating System/Batch Environment (NOS/BE) Version 1.5, NOS/BE can operate on 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 and 875

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

CDC 6000 Computer Systems

This handbook describes the general installation process, lists the operating and product set release materials, and gives detailed procedures and installation parameters needed for the operating system and individual product set members.

AUDIENCE

This handbook is written for the systems analyst who is familiar with the COMPASS assembly language, the SYMPL programming language, the hardware configuration on which NOS/BE is installed, and the NOS/BE commands.

ORGANIZATION

This manual is divided into three parts.

Part I	Lists the installation and verification job decks and the options available with them; it also outlines the general installation procedure and illustrates the order of installation.
Part II	Lists the release materials, detailed installation procedures and parameters, and any additional information for the operating system and each product set member.
Part III	Contains a cross-reference listing showing routines that reference installation parameters.

CONVENTIONS

CYBER 70 COMPUTER SYSTEMS

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

170-CLASS MAINFRAMES

The term 170-class mainframes refers to CYBER 170 models 171, 172, 173, 174, 175, 176, 720, 730, 740, 750, 760, 865 and 875.

180-CLASS MAINFRAMES

The term 180-class mainframes referes to CYBER 180 models as well as CYBER 170 models 815, 825, 835 and 855, which have most of the functional and architectural attributes of the CYBER 180 mainframes.

EXTENDED MEMORY

Extended memory for model 176 is a large central memory extended (LCME). Extended memory for the 180-class mainframes is unified extended memory (UEM).

Extended memory for models 865 and 875 can be UEM, extended core storage (ECS), or extended semiconductor memory (ESM). Extended memory for all other NOS/BE computer systems is ECS or ESM. ECS and ESM in 24-bit format standard addressing mode (sometimes called ECS mode) are the only forms of extended memory that can be shared in a multimainframe complex, can be accessed by a distributive data path (DDP), and can be used as station link media. In this manual, ECS refers to both ECS and ESM, and extended memory refers to all forms of extended memory.

SUBMITTING COMMENTS

The last page of this manual is a comment sheet. Please use this comment sheet to give us your opinion of the manual's usability, to suggest specific improvements, and to report technical or typographical errors. If the comment sheet has already been used, you can mail your comments to:

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

Please include the manual title, publication number, and revision level with each inquiry, and indicate whether or not you would like to reply.

Additionally, if you have access to SOLVER, an online facility for reporting problems, you can use it to submit comments about the manual. Use NBO as the product identifier.

RELATED PUBLICATIONS

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. Hardware descriptions of the various forms of extended memory can be found in the following 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

The NOS/BE Manual Abstracts is a pocket-sized manual containing brief descriptions of the contents and intended audience of all NOS/BE and NOS/BE product manuals. The abstracts can be useful in determining which manuals are of greatest interest to a particular user.

The following is a list of NOS/BE operating system manuals, NOS/BE product set reference manuals, and other manuals of interest.

Manuals, except for the NOS/BE System Programmer's Reference Manual, Volumes 1 and 2, are available through Control Data sales offices or Control Data Literature Distribution Services (308 North Dale Street, St. Paul, Minnesota 55103). The NOS/BE System Programmer's Reference Manual, Volumes 1 and 2, is available through Software Manufacturing Distribution (ARH230, 4201 North Lexington Avenue, St. Paul, Minnesota 55112).

Control Data Publication	Publication Number
BASIC Version 3 Reference Manual	19983900
COBOL Version 5 Diagnostic Handbook	60482500
COBOL Version 5 Reference Manual	60497100
Common Memory Manager Version 1 Reference Manual	60499200
Communications Control INTERCOM Version 3 Reference Manual	60471150
COMPASS Version 3 Reference Manual	60492600
Concurrent Maintenance Library Version 3.2 Reference Manual	60455980
CYBER Cross Build Utilities Reference Manual	60471200
CYBER Cross System 1 Diagnostic Handbook	96836300
CYBER Cross System 1 Macro Assembler Reference Manual	96836500
CYBER Cross System 1 Micro Assembler Reference Manual	96836400
CYBER Cross System 1 PASCAL Compiler Reference Manual	96836100
CYBER Cross System 1 Reference Manual	96836000
CYBER Database Control System Version 2 Reference Manual	60481800

Control Data Publication	Publication Number
CYBER Initialization Package (CIP) User's Handbook	60457180
CYBER Interactive Debug Version 1 Reference Manual	60481400
CYBER Loader Version 1 Reference Manual	60429800
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300
CYBER Record Manager Basic Access Methods Version 1 Reference Manual	60495700
CYBER Record Manager Version 1 Guide for Users of FORTRAN Extended Version 4	60495900
CYBER 170 Models 835, 845, and 855 Hardware Reference Manual	60469290
CYBER 170 Models 815 and 825 Hardware Reference Manual	60469370
CYBER 180 Models 810 and 830 Hardware Reference Manual	60469420
Data Base Utilities Version 1 Reference Manual	60498800
Data Catalogue 2 Version 1 Reference Manual	60483200
Data Handler Version 1 Reference Manual	17322100
DDL Version 3 Reference Manual	
Volume 1 Schema Definition	60481900
Volume 2 COBOL Sub-Schema Definition	60482000
Volume 3 QUERY UPDATE Sub-Schema Definition	60482100
EXPORT High Speed Reference Manual	60456880
FORM Version 1 Reference Manual	60496200
FORTRAN Common Library Mathematical Routines Reference Manual	60498200
FORTRAN Data Base Facility Reference Manual	60482200
FORTRAN Extended Version 4 Debug User's Guide	60498000
FORTRAN Extended Version 4 Reference Manual	60497800
FORTRAN Version 5 Reference Manual	60481300
FORTRAN 4 to 5 Conversion Aid Reference Manual	60483000
INTERCOM Version 5 Guide for Users of COBOL	60455960
INTERCOM Version 5 Guide for Users of FORTRAN Extended	60455950
INTERCOM Version 5 Interactive Command Summary	60455840

Control Data Publication	Publication Number
INTERCOM Version 5 Multiuser Job Capability Reference Manual	60456070
INTERCOM Version 5 Reference Manual	60455010
INTERCOM Version 5 Remote Batch Command Summary	60455850
INTERCOM Version 5 Remote Batch User's Guide	60455890
MSL 15X Off-Line Maintenance Software Library Reference Manual	60456530
MSL 140 Off-Line Maintenance Software Library Reference Manual	60459860
NOS/BE Version 1 Applications Installation Handbook	84000980
NOS/BE Version 1 Diagnostic Handbook	60494400
NOS/BE Version 1 Diagnostic Index	60456490
NOS/BE Version 1 On-Line Maintenance Software Reference Manual	60453900
NOS/BE Version 1 Operator's Guide	60493900
NOS/BE Version 1 Reference Manual	60493800
NOS/BE Version 1 System Programmer's Reference Manual, Vol. 1	60494100
NOS/BE Version 1 System Programmer's Reference Manual, Vol. 2	60458490
NOS/BE Version 1 User's Guide	60494000
Programming Reference Aids	60158600
Query Update Version 3 Reference Manual	60498300
Remote Host Facility Access Method Reference Manual	60459990
Remote Host Facility Usage	60460620
SCED Version 1 User's Guide	60494800
SCOPE 2.1 Operator's Guide	60455090
SCOPE 3.3 to SCOPE 3.4 Conversion Aids Programming Systems Bulletin	60493200
SIFT Version 1 Programming Systems Bulletin	60496500
Sort/Merge Version 5 Reference Manual	60484800
SYMPL Version 1 Reference Manual	60496400
Update Version 1 Reference Manual	60449900
8-Bit Subroutines Reference Manual	60495500
380-170 Network Access Device Hardware Reference Manual	60458500
1700 MSOS IMPORT HS 1 Reference Manual	96719200

The NOS/BE to NOS/BE link is described in the NOS/BE Version 1 Operator's Guide. The NOS/BE to SCOPE 2 link is described in the SCOPE 2 Operator's Guide.

Effective with the release of PSR Level 499 (NOS/BE 1.3, third corrective code release), CEMS support for the 657 and 659 tape units and 841 multiple disk drive is withdrawn. References to these devices and their interfaces to the software have been deleted from all product support manuals except this manual. Although the installation information for these devices is retained in this manual, it is not to be construed as continued CEMS support.

Rather, it is an accommodation to your continued use of the equipment and the code supporting this equipment, despite the withdrawal of future CEMS support.

DISCLAIMER

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

CONTENTS

PART I

1. GENERAL INSTALLATION I-1-1

NOS/BE Installation Process I-1-1
User Library Method I-1-1
Running System Modification Method I-1-1
Build Preparation I-1-2
Build Techniques I-1-5
User Library Techniques I-1-5
Complete Builds and Assemblies I-1-5
Partial Builds and EDITLIBs I-1-6
Deadstart Tape Creation I-1-6
Running System Modification Techniques I-1-6
Deadstart Tape Creation and Text Record Use I-1-7
Product Dependencies I-1-8
Installation Deck Nomenclature I-1-13
AUTOMAT Procedure I-1-14
Obtaining Installation Jobs I-1-14
Deck Options I-1-16
Common Deck Modifications I-1-19
DST1 and DST2 Common Decks I-1-20
Special Purpose Decks I-1-21
MINI I-1-21
ULIB I-1-21
LIBS I-1-21
DST1 I-1-21
DST2 I-1-22
DST3 I-1-22
IPTEXT I-1-22
COMPCOM I-1-22
PL1AT I-1-23
System Texts I-1-23

PART II

1. NOS/BE VERSION 1.5 II-1-1

Hardware Configuration II-1-1
Release Materials II-1-1
CIP Install Tape II-1-3
Notes and Cautions II-1-4
Notes and Cautions (Shared RMS Environment) II-1-6
Notes and Cautions (Tape Scheduling) II-1-6
Notes and Cautions (ATS 679 Tape Drives) II-1-6
Installation Procedures II-1-7
General Installation Parameters (IPARAMS) II-1-8
Tape Processing Installation Parameters (CIOCOM) II-1-18
Automatic Assignment Bit II-1-21
Automatic Scratch Status II-1-21
Prestaging Bits II-1-21
Unit Reservation II-1-21
Prestaging II-1-21
Overcommitment Bit II-1-22
Miscellaneous Bits II-1-22
Two-Label Processors II-1-22
EOJ Tape Unload II-1-22
Operator Cannot Override VSN II-1-22
Label Debug II-1-22
Option Dependencies (IP.TSG) II-1-23
Auto Assign Dependencies II-1-23
Prestage and Overcommitment Dependencies II-1-23
Tape Security Dependencies II-1-23
IP.WEC II-1-23
Miscellaneous Installation Variables II-1-24
Password Definitions for the System Dayfile II-1-24
Macros to Define Spacing Code Arrays for Janus II-1-24
Macro to Define the EC Parameter on the Job Statement II-1-25
CMR Configuration Parameters (CMRIP) II-1-26
Logical ID Table (LID) II-1-30
System Seconds II-1-31
Weighting Constants II-1-31
Table Structures II-1-32
Equipment Status Table II-1-32
Equipment Configuration II-1-33
Record Block Reservation Table II-1-41
FLAW Table II-1-46
Computing Physical Addresses for Converting RB Numbers II-1-47
CMR Equipment Configuration Examples II-1-49
Deadstart Installation Parameters (DSLCOM) II-1-51
Disk Preparation II-1-54
Permanent Files and Device Set Installation II-1-63
Disk Space Threshold Setting II-1-66

Scheduling Parameters II-1-69
Job Classes II-1-69
Central Memory II-1-75
Extended Memory Use II-1-76
CPU II-1-77
System Idle Mode II-1-77
Extended Memory Installation Parameters (ECSCOM) II-1-78
System Circular Buffers (SCB) II-1-79
Macros to Construct Extended Memory Label II-1-80
ECS System Segments II-1-82
Segment Activity Count II-1-82
Calling Sequence II-1-83
ECS as a Link Medium Between Mainframes II-1-84
Update Installation Options II-1-85
Common Memory Manager Version 1 (CMM) II-1-85
Deadstart Loading the Operating System II-1-86
Controlware Binary Creation II-1-87
Coldstart Deck Creation II-1-87
Warmstart Deck Creation II-1-87
Sample Job for Creation of Installation CMRs and CMR Libraries II-1-89
System Residency II-1-90
RHF Support; Logfile Defintion II-1-93
Accounting for QLOG II-1-94
Station Support; SPOTS Core Requirements II-1-94
Accounting for the SPOTS Station II-1-96
Accounting for GENLDPF-Generated Jobs II-1-103
Gemini Load Leveling II-1-103
Installation Parameters II-1-104
Load-Leveling Algorithm II-1-105
Notes and Cautions II-1-105
Deadstart Dump Analyzer II-1-106
Accounting for Controlware Dump Files II-1-106

2. COMPASS VERSION 3.6 AND COMMON COMMON DECKS II-2-1

Release Description II-2-1
Release Materials II-2-1
General Description II-2-1
Installation Parameter II-2-1
Installation Procedures II-2-2

3. CYBER RECORD MANAGER 1 BASIC ACCESS METHODS II-3-1

Release Description II-3-1
Notes and Cautions II-3-1
Additional Information II-3-1
Initialization Modules II-3-1
Sequential and Word Addressable I/O Modules II-3-1
Installation Parameters II-3-3
Installation Procedures II-3-3

4. CYBER RECORD MANAGER 2 ADVANCED ACCESS METHODS II-4-1

Release Description II-4-1
Release Materials II-4-1
Limitations II-4-1
AAM Installation Parameters II-4-2
User Additions to AAM II-4-2
Installation Procedures for AAM II-4-5
Verification Programs II-4-5

5. 8-BIT SUBROUTINES VERSION 1.1 II-5-1

Release Description II-5-1
Release Materials II-5-1
Hardware Configuration II-5-1
General Description II-5-1
Installation Procedures II-5-1

6. FORM VERSION 1.1 II-6-1

General Description II-6-1
Release Materials II-6-1
Hardware Configuration II-6-1
Installation Parameters II-6-1

7. ON-LINE MAINTENANCE SOFTWARE II-7-1

Release Description II-7-1
Release Materials II-7-1
Installing Diagnostic Programs II-7-1
Installation Parameters II-7-2
Installation Procedures II-7-2

8. MAINTENANCE PACKAGE II-8-1

Release Description II-8-1
SYMPL II-8-1
Installation Parameters II-8-1
Installation Procedures II-8-1
Verification Program II-8-2
Conversion Aids II-8-2
Installation Procedures II-8-2
Use of Conversion Aids II-8-2
SPY Use II-8-3
Output II-8-3
Dual CPU II-8-4
Swapping II-8-4
Checkpoint II-8-5
Calling SPY II-8-5
Terminating SPY II-8-7
PRNTSPY II-8-7
DOCK Use II-8-8
SIS63 Use II-8-9
CIA Use II-8-9
Percentages II-8-10
CPMET Use II-8-11

9. FORTRAN EXTENDED VERSION 4 II-9-1

Release Materials II-9-1
Limitations II-9-1
Compiler Installation Parameters II-9-1
Compiler Program Library Structure II-9-2
TSTEXT II-9-3
FTNMAC II-9-3
FTNTEXT II-9-3
TS Mode Overlay II-9-4
Optimizing and Debugging Mode Overlays II-9-4
(2,0) Overlay (FTN20) II-9-4
(2,3) Overlay (FTN23) II-9-4
(2,1) Overlay (FTN21) II-9-4
(2,2) Overlay (FTN22) II-9-5
(2,5) Overlay (FTN25) II-9-5
(2,4) Overlay (FTN24) II-9-5
Installation Instructions II-9-5

10. COBOL VERSION 4.7 II-10-1

This section has been removed.

11. SORT/MERGE 5 II-11-1

Release Materials II-11-1
Hardware Configuration II-11-1
Installation Parameters II-11-1
Installation Procedure II-11-1

12. INTERCOM VERSION 4 II-12-1

This section has been removed.

13. ALGOL 60 VERSION 4 II-13-1

This section has been removed.

14. REMOTE DIAGNOSTIC FACILITY II-14-1

Release Materials II-14-1
Installation Parameters II-14-1
Equipment Status Table II-14-1
Installation Procedures II-14-1
RDF Use of PASSWRD II-14-1

15. CYBER CROSS SYSTEM VERSION 1 II-15-1

Release Description II-15-1
Hardware Configuration II-15-1
Release Materials II-15-2
Pascal Organization II-15-3
Pascal Compiler Symbol Table II-15-3
Installation Procedure II-15-3

16. CYBER LOADER 1 II-16-1

Release Description II-16-1
Release Materials II-16-1
General Description II-16-1
Installation Parameters II-16-2
Installation Procedures II-16-3

17. FACTORY FORMAT SUPPORT (844-21 AND 844-41) II-17-1

Release Materials II-17-1
General and Operational Description II-17-1
Flaws II-17-1
Disk Pack Condition II-17-2
Installation Information II-17-3
Format Utility II-17-3
Call Format and Parameter Options II-17-3
Input Formats II-17-4
Output Formats II-17-5
Operator Intervention and Console Messages II-17-5
Dayfile Messages II-17-6
Installation Procedures II-17-7

18. CYBER DATABASE CONTROL SYSTEM 1.2 II-18-1

This section has been removed.

19. QUERY UPDATE VERSION 3 II-19-1

Release Materials II-19-1
Hardware Configuration II-19-1
Installation Parameters II-19-1
Notes and Cautions II-19-1
Installation Requirements II-19-2
Installation Procedure II-19-2

20. DATA DESCRIPTION LANGUAGE VERSION 2.2 II-20-1

This section has been removed.

21. BASIC VERSION 3 II-21-1

Release Materials II-21-1
Hardware Configuration II-21-1
Notes and Cautions II-21-1
Installation Parameters II-21-1
Installation Procedures II-21-2
Verification Program II-21-2

22. DATA BASE UTILITIES 1.2 II-22-1

Release Description II-22-1
Release Materials II-22-1
Installation Procedure II-22-1
Limitations II-22-1

23. COMMUNICATIONS CONTROL PROGRAM 1.1 II-23-1

This section has been removed.

24. COBOL 5 II-24-1

Release Materials II-24-1
Limitations II-24-1
Installation Parameters II-24-1
Compiler Program Library Structure II-24-3
Installation Instructions II-24-4

25. COBOL 4 TO COBOL 5 CONVERSION AID II-25-1

Release Materials II-25-1
Hardware Configuration II-25-1
General Description II-25-1
Installation Procedure II-25-1
Installation Parameters II-25-2

26. CYBER CONTROL LANGUAGE II-26-1

Release Materials II-26-1
Hardware Configuration II-26-1
General Description II-26-1
Notes and Cautions II-26-1
Installation Parameters II-26-1
Installation Procedure II-26-3

27. INSTALLATION PROCEDURE FOR MODEL 171 CONFIGURATION II-27-1

Required Materials II-27-1
Restrictions II-27-1
Required Macros II-27-1
  AUTO Macro II-27-2
  MUXCR Macro II-27-2
Installation Procedures for INTERCOM 5 II-27-5

28. INTERCOM VERSION 5 II-28-1

Release Description II-28-1
Hardware Configuration II-28-1
  Required Hardware Options II-28-1
  Hardware Options II-28-2
Release Materials II-28-2
Notes and Cautions II-28-2
  Line Skipping on 714 Nonimpact Printer II-28-2
  Character Set Support II-28-3
  Limitations and System Considerations II-28-3
  General Procedures II-28-3
Installation Parameters II-28-3
INTERCOM Common Deck Settings II-28-4
NOS/BE IPARAMS Settings II-28-8
EST Entry II-28-9
Multiplexer Tables II-28-9
  COM2CC Macro II-28-10
  MUJ Macro II-28-10
  EXPCOM Macro II-28-10
  REMOTE Macro II-28-11
MUJ Table Structure (1QP) II-28-11

TBL Assembly Options	II-28-11
EDITOR Installation Parameters	II-28-12
EDITOR Debug Code	II-28-21
Multiuser Job Installation Parameters	II-28-21
Installation Procedures	II-28-22
NPU Programs	II-28-23
MUJ System Errors	II-28-23
Multiplexer Configuration	II-28-24
Multiplexer Definition Statement	II-28-25
Port Definition Statement	II-28-25
Port Definition Parameters	II-28-25
DEBUG Statement	II-28-29
Command Table Structure (1CI Overlay 2CS - Common Deck COMTBL)	II-28-29
Password File Creation	II-28-30
SCED Installation Parameters	II-28-34
Verification Procedure	II-28-34.1

29. COMMUNICATIONS CONTROL INTERCOM 3	II-29-1
Release Description	II-29-1
Installation Overview	II-29-2
Release Materials	II-29-3
Hardware Configuration	II-29-4
CCI Hardware Requirements	II-29-4
Notes and Cautions	II-29-5
Installation Parameters	II-29-5
Type 1 Statements	II-29-5
Type 2 Statements	II-29-6
Type 3 Statements	II-29-8
Configuration Aids	II-29-8
Installation Procedures	II-29-9
2550-2 or 2551-1 Installation (PL99A)	II-29-9
Installation Deck PL99E	II-29-9
Corrective Code	II-29-9
Verification Programs	II-29-10
System Generation	II-29-10
On-Line System	II-29-10

30. CYBER DATABASE CONTROL SYSTEM 2	II-30-1
Hardware Configuration	II-30-1
Release Materials	II-30-1
Installation Requirements	II-30-1
Installation Procedure	II-30-2
Verification Programs	II-30-2
CDCS Accounting Table	II-30-2

31. DATA DESCRIPTION LANGUAGE VERSION 3	II-31-1
Release Materials	II-31-1
Hardware Configuration	II-31-1
Installation Procedure	II-31-1

32. CYBER INTERACTIVE DEBUG 1	II-32-1
Hardware Configuration	II-32-1
Release Materials	II-32-1
Installation Parameters	II-32-1
Installation Procedures	II-32-2

33. PL/I 1	II-33-1

This section has been removed.

34. FORTRAN COMMON LIBRARY VERSION 4 WITH POSTMORTEM DUMP UTILITY	II-34-1
Release Materials	II-34-1
Installation Options	II-34-1
FCL Program Library Structure	II-34-1
PMD Program Library Structure	II-34-2
Installation Decks	II-34-2

35. FORTRAN DATABASE FACILITY VERSION 1	II-35-1
Release Materials	II-35-1
Hardware Configuration	II-35-1
Notes and Cautions	II-35-1
Installation Procedure	II-35-1
Verification Program	II-35-1

36. ALGOL 60 VERSION 5	II-36-1

This section has been removed.

37. FORTRAN 4 TO FORTRAN 5 CONVERSION AID II-37-1

Release Materials II-37-1
Hardware Requirements II-37-1
Installation Procedures II-37-1
Verification Program II-37-1

38. FORTRAN VERSION 5 II-38-1

Release Materials II-38-1
Limitations II-38-1
Installation Parameters II-38-1
Installation Procedure II-38-2

39. FORTRAN COMMON LIBRARY VERSION 5 WITH POSTMORTEM DUMP UTILITY II-39-1

Release Materials II-39-1
Installation Options II-39-1
FCL5 Program Library Structure II-39-1
PMD Program Library Structure II-39-1
Installation Procedures II-39-2

40. EXPORT HIGH SPEED II-40-1

Release Materials II-40-1
Hardware Configuration II-40-1
Limitations II-40-1
Installation Parameters II-40-2
Equipment Status Table II-40-2
Installation Procedures II-40-2
EXPORT Use of Terminal IDs II-40-2

41. DATA CATALOG 2 VERSION 1 II-41-1

Release Materials II-41-1
Installation Procedure II-41-1

42. REMOTE HOST FACILITY (RHF) II-42-1

Release Materials II-42-1
Installation II-42-1
Hardware Configuration II-42-2
NAD Controlware Loading II-42-2
Non-170 NAD Controlware Installation II-42-4
380-370 NAD Controlware Installation (IBM) II-42-4
380-110 NAD Controlware Installation (MIN) II-42-5
NAD Controlware Initialization Parameters II-42-5
Installation Parameters II-42-6
Application Accounting Definition II-42-8
FTFS Installation Parameters II-42-8
Maintenance Host Facility (MHF) II-42-9
MFLINK Delay File II-42-10
RHF Configuration File Generation II-42-11
Network Configuration Statements II-42-13
Remote Mainframe Definition II-42-13
LID Definitions for the Remote Mainframe II-42-13
Path Definitions to the Remote Mainframe II-42-14
Remote NAD Definitions II-42-14
Local NAD Definitions II-42-15
Logical Identifier Definition and Use II-42-18
Special Loopback Capability II-42-19
Application Definitions II-42-20
Definition of Maximum NDRs Allowed for All NADs II-42-21
DEBUG Parameter Definition II-42-22
UCP Charge Definition for an RHF Call II-42-22
Verification Program II-42-22
QTF Configuration Requirements II-42-23
RHF Configuration File Directives II-42-23
QTF Procedure File II-42-23
Procedure File Example II-42-24

43. INTERACTIVE TRANSFER FACILITY (ITF) II-43-1

Release Materials II-43-1
Hardware Configuration II-43-1
Installation Procedure II-43-1
Installation Parameters II-43-1
Verification Procedure II-43-3

44. NOS/VE DUAL-STATE PREPARATION II-44-1

Software Requirements II-44-1
Building NOS/VE Dual-State Binaries on NOS/BE II-44-1
Step 1, Establish a NOS/VE Installation User Name II-44-2
Step 2, Generate NOS/BE Configuration Changes II-44-2
NVE Subsystem Support II-44-2
EST Definitions II-44-2
CM Allocation II-44-3
Step 3, Install the NOS/VE Release Files II-44-3

Step 4, Create the NOS/BE Deadstart File II-44-4
Step 5, Redeadstart NOS/BE II-44-4
Step 6, Add Information for Partner Jobs II-44-5
Step 7, Validate Users on NOS/BE II-44-7

PART III

1. CROSS-REFERENCE OF INSTALLATION PARAMETERS III-1-1

APPENDIX

A. GLOSSARY A-1

INDEX

FIGURES

I-1-1 Installation Sequence I-1-9
II-1-2 Batch/INTERCOM Considerations II-1-70
II-8-1 CP Monitor Execution Timing Report II-8-12
II-28-1 2CS Release Values and Update Identifiers II-28-12
II-28-2 1QP Release Values and Update Identifiers II-28-13
II-42-1 Sample LCN Network II-42-16

TABLES

I-1-1 Installation Requirements I-1-10
I-1-2 Common Decks and System Texts I-1-24
I-1-3 Product Installation I-1-25
II-1-1 RMS Device II-1-43
II-1-2 Disk File Allocation II-1-43
II-1-3 Core Requirements for Various SPOTS II-1-95
II-28-1 EDITOR Installation Parameters II-28-16
II-28-2 EDITOR Dependencies II-28-20
II-28-3 EDITOR Array Dimensions in IPFTN II-28-21
II-29-1 CCI Module Requirements II-29-1

PART I

GENERAL INSTALLATION 1

NOS/BE INSTALLATION PROCESS

Installation of the NOS/BE system is an incremental process in which you enter each product binary file into either a set of user library permanent files or into the host NOS/BE operating system by using the EDITLIB utility. This is an ordered process in which you must maintain strict adherence to product interdependencies. Several build options allow you both to integrate the products supported by NOS/BE and to tailor these products on site. Two generalized methods for producing an integrated, tailored system, the user library method and the running system modification method, are provided.

USER LIBRARY METHOD

The user library method is the most reliable and efficient method for producing an integrated, tailored NOS/BE system. This installation method maintains each product binary file in a set of user library permanent files on either a private or public device set. This set of user libraries is self-contained, that is, the host NOS/BE operating system supplies only the basic operational capabilities of file manipulation, job scheduling, and resource management.

The user library method has the following advantages:

- It does not interrupt normal batch and interactive processing in a production environment.
- It provides more reliable recovery from possible failures of the host NOS/BE system.

Disadvantages of this method are:

- You do not discover certain types of build errors until you create a deadstart tape.
- Some host system modifications may be required when a key interface or a build tool such as EDITLIB changes.

This method is used by NOS/BE development to build all intermediate development systems.

RUNNING SYSTEM MODIFICATION METHOD

With the running system modification method, you enter each product binary file into the host NOS/BE system either from a release tape or from the reassembled binaries produced by a previously executed assembly deck.

Advantages of this method are:

- It enables you to verify the integrity of these new binaries by executing verification programs as soon as the new binaries are installed.
- It conserves mass storage space.

Disadvantages of this method are:

- A production environment cannot be maintained during the build process without frequent interruptions.

- Recovery of the modified NOS/BE host system may be impossible.

BUILD PREPARATION

Installation of a NOS/BE system requires that you have a working knowledge of the utilities Update and EDITLIB. Familiarity with the operational characteristics of the NOS/BE operating system, its constituent texts, and control language statements is assumed. Additionally, knowledge of the functional characteristics of COMPASS, CYBER Loader, COPYL, ITEMIZE, FORTRAN Extended 4, FORTRAN Common Library 4, SYMPL, and CYBER Control Language is required for installation of the basic operating system components. Refer to later sections of this manual for dependencies that require knowledge of other system components.

In addition to the information contained in this document, supplemental information related to a specific release level is contained in the Software Release Bulletin associated with that release level. Deck REASON on the installation decks old program library (oldpl) also contains specific release installation information. The installation decks oldpl is present as file 3 of the batched corrective code (BCC) tape.

The BCC tape, which is supplied with the release, has the following structure.

File	Description
1 and 2	Critical code for the NOS/BE system. File 1 contains code for the operating system, and file 2 contains code for the product set members. The code on files 1 and 2 are Update program libraries that use a slash (/) as the master control character. Update decks include /CALL directives for relevant common decks. To extract the critical code for a specific product, use a /COMPILE directive for that deck. Deck names are listed in table I-1-3 later in this section.
3	Installation and verification job deck Update program library that uses the equal sign (=) as the master control character. This file contains various options embedded in the job deck. Execute the job sequence described later in this section to obtain definitions of these options and a list of the decks affected.
4	PSR index pertaining to the operating system and product set. Record 1 — Contains the PSRs published since the last release sorted by: Operating system/product set Site Product Record 2 — Contains the PSRs published since the last release sorted by: Operating system/product set Decks/routines affected Product

File		Description
	Record 3	Contains the PSRs published since the last release sorted by: Operating system/product set Product Site
5		Update oldpl, using a slash as the master control character and contain source for suggested code. Read deck REASON (contained in the installation decks oldpl) concerning the use of this file.
6		Critical code for RHF. The code on file 6 is an Update program library that uses a slash (/) as the master control character. Update decks include /CALL directives for relevant common decks. To extract the critical code for a specific product, use a /COMPILE directive for that deck. Deck names are listed in table I-1-3 later in this section.

Part III of this document contains a cross-reference listing to aid in the examination of generalized installation parameters that may be tailored to a specific configuration.

Submit the following job both to obtain a listing of the installation decks oldpl for use as a reference during the installation process and to save these decks as a permanent file for subsequent use. The output from this job will be a deck list itemizing the installation jobs and various utility decks on file 3 of the corrective code tape. Deck REASON is one of the decks that will be listed; it contains a discussion of the various options embedded within those decks.

Job	Comment
DECKS,T100,MT1. (or PE1),(or HD1)	
REQUEST(OLDPL,HY)	Assign BCC tape.
REQUEST(NEWPL,*PF)	
SKIPF(OLDPL,2,17)	
UPDATE(F,N,*==,C=0,R,L=A7)	
CATALOG(NEWPL,DECKS,ID=INSTALL)	
6/7/8/9	

To select an installation deck from the installation decks oldpl, execute the following job.

```
AUTO,TO.
ATTACH(OLDPL,DECKS,ID=INSTALL)
REQUEST(COMPILE,*Q)
UPDATE(*==,Q,D,8)
ROUTE(COMPILE,DC=IN)
7/8/9
=DEFINE x
=C y
6/7/8/9
```

x is an option or combination of options described in the listing of deck REASON (obtained from the list of the installation jobs). y is a deck name contained on this program library (PL).

Punch installation decks by executing the following job.

```
DECKS,TO.
ATTACH(OLDPL,DECKS,ID=INSTALL)
UPDATE(*==,Q,D,8,C=PUNCH)
7/8/9
=DEFINE x
=C y
6/7/8/9
```

x is one or more of the options available on the installation decks file. y is one or more of the job decks on the same file.

The program library on file 5 of the BCC tape contains a deck named DOLLAR. The FORTRAN program in this deck can be used to list the code identification, history information, card count, routines modified from file 5, or the delta code (source code of PSRs from previous level to this level) from an operating system PL. Use the following job form to obtain this information.

Job	Comment
JOB,TO,PE1.	
REQUEST(BCC,PE)	Assign BCC tape.
SKIPF(BCC,4,17)	Position to file 5.
COPYBF(BCC,OLDPL)	
UPDATE(*=/,Q,D)	
UNLOAD(BCC)	
FTN(I=COMPILE)	
UPDATE(P=nnn,F,*=/,S=TAPE5,C=TAPE1)	nnn=OLDPL, an operating system program library, or an operating system delta code program library.
LGO.	
REWIND(TAPE2)	
COPYCF(TAPE2)	
7/8/9	
/C DOLLAR	
7/8/9	
//GENERATE FULL COMPILE FILE-OPERATING SYSTEM CODE	
7/8/9	
data card	
6/7/8/9	

The data card places page heading information on TAPE 2 and should be in the following form.

columns 1 through 8 date

columns 11 through 49 page title

Execute the following job to obtain a listing of the lower CYBER PSR index described earlier in this section.

Job	Comment
JOB,TO,PE1.	
REQUEST(OLDPL,PE)	Assign BCC tape.
SKIPF(OLDPL,3,17)	Skip to fourth file.
COPYBR(OLDPL,OUTPUT)	List numerical sort transaction log.
COPYBR(OLDPL,OUTPUT)	List site code sort transaction log.
COPYBR(OLDPL,OUTPUT)	List full index sorted on routine.
UNLOAD(OLDPL)	
6/7/8/9	

BUILD TECHNIQUES

The key deck in either the user library or running system method of building systems is the deadstart creation deck. Use DST1 for the user library method; use DST2 for the running system method. Inspect the conditional UPDATE =IF DEF directives in DST1 and DST2, and the source statements they govern, for information regarding the eventual content and library structure of the deadstart tape to be created. Deck REASON provides a shortened list of these definable parameters as well as all definable parameters for other decks. From these references, create a generalized set of =DEFINE UPDATE directives that reflect all of the parameters associated with the products to be installed. Each deck extracted from the installation decks oldpl should be extracted with this generalized set of defines.

The installation decks call other UPDATE common decks, which provide one location for installation deck modifications referenced more than once. This implementation is generalized as much as possible to provide a basis for the CYBER Control Language procedure file to be used for deck extraction. (Examples of this use are contained in deck REASON.) Specialized common decks, which provide locations in nonrepeatable sequences, such as in DST1 and DST2, are documented later in this section.

USER LIBRARY TECHNIQUES

The deck LIBS establishes the permanent file environment for the user library process. All user libraries are created with a dummy routine ZZZ in each library. Subsequent installation decks delete this dummy routine. Add the UPDATE directive =DEFINE ULIB to the generalized set of defines being used when installing with the user library method.

Complete Builds and Assemblies

Extract all appropriate I-suffixed decks and submit for execution. You must honor all product dependencies.

Partial Builds and EDITLIBs

You can enter any product for which there is no corrective code into the user libraries by first executing the appropriate O-suffixed deck, if one exists. (Each product generating relocatable binary modules that must be changed to absolute binary overlays should have a corresponding O-suffixed deck.) Assign the output tape from the O deck to the corresponding E-suffixed deck to enter the new binaries into the user libraries. For products containing no overlays, the E-suffixed deck is the only required deck. The NOS/BE system contained on PL1A and PL1B, as well as INTERCOM 5, require that the I deck be run. Some products provide special partial assembly or variant decks. Carefully examine and understand these decks before using them.

As in complete builds, you must honor all product dependencies.

Deadstart Tape Creation

Upon completion of execution of the last product installation deck, save all user libraries on tape using a mode 1 DUMPF. Before executing the DST1 deadstart tape creation deck, you must execute the utility deck ULIB. This deck creates a sequential library of the libraries USERPP and USERPS. It merges user library USERCC with library USERNUC and invalidates USERNUC for subsequent rebuilds, if necessary. (You can use the SAVE define to create a second cycle of USERNUC and avoid invalidating the original library. If you use this option, purge the high cycles of USERPP, USERPS, and USERNUC before any rebuilding.)

Several alternative file locations for controlware, CMR libraries, CMRs, and diagnostic sequencer text records exist in DST1. These options, along with alternatives for system residency, require that DST1 be tailored to meet specific needs.

RUNNING SYSTEM MODIFICATION TECHNIQUES

After each appropriate I-suffixed deck has completed execution, execute the corresponding E-suffixed deck. Some products do not have a corresponding E deck; for these products, add the binaries to the new system by the DST2 deadstart tape creation deck. Where an E deck exists, assign the output tape from each I deck as input to the corresponding E deck. You must honor all product dependencies, including completion of the E decks, before installing dependent products. A few intermediate system EDITLIBs occur during execution of the I decks, primarily to upgrade system text files. Avoid partial builds using the O decks since the frequency of their use is rather low on system levels different from the system level being built. This is true for partial assembly and variant decks also.

DEADSTART TAPE CREATION AND TEXT RECORD USE

An example of how to capture the host NOS/BE system and its constituent libraries follows.

Job	Comments
CAPTURE,I00,T0,PE1. (Or HD1;MT1 for seven-track)	
ACCOUNT(local accounting information)	
REQUEST(NEWSYS,PE,RING,VSN=1234)	
EDITLIB(SYSTEM,ERROR=3,MSGL=1)	
UNLOAD(NEWSYS)	
REWIND(OUTPUT)	
7/8/9	
READY(NEWSYS,NEW)	
REWIND(NEWSYS)	
TRANS77(IPL+ZZZ,SYSTEM)	
TRANS77(OSB,SYSTEM)	CTI records are optional; record ZZZ is required.
TRANSFER(CED+MDR,SYSTEM)	
TRANSFER(1,SYSTEM)	Assuming only one CMR is present.
TRANSFER(COM+LFP,SYSTEM)	
TRANSFER(1,SYSTEM)	Applicable for 819 driver only; optional.
TRANSFER(OSY,SYSTEM)	844 half-tracking controlware; optional.
TRANSFER(OSZ,SYSTEM)	844 full-tracking controlware; optional.
TRANSFER(OMT,SYSTEM)	MTS controlware for 66x tapes; optional.
TRANSFER(OSJ,SYSTEM)	885 disk drive controlware; optional.
TRANSFER(*,SYSTEM)	This directive is invalid if an attempt is made to omit an optional host system controlware package. Some combination of the preceding optional controlware is required by most sites.
INCLUDEP(SYSTEM)	An INCLUDE directive is required for each system library to be present on the new deadstart tape.
INCLUDE(NUCLEUS,SYSTEM,CM)	
INCLUDE(SYSOVL,SYSTEM,CM)	
INCLUDE(BAMLIB,SYSTEM,DS)	
INCLUDE(SYMLIB,SYSTEM,DS)	
INCLUDE(FORTRAN,SYSTEM,DS)	
.	
.	
.	
COMPLETE.	
ENDRUN.	
7/8/9	
6/7/8/9	

Either an EDITLIB LISTLNT of the host NOS/BE system or an ITEMIZE listing of the host deadstart tape is required to construct the correct sequence described. Additionally, determine the controlware packages and optional product libraries required for the target machine configuration. An examination of the deadstart tape creation decks supplied on the installation decks oldpl can serve as a guide to the creation of this capture deck. A simplified capture deck, named DST3, is provided on the installation decks oldpl.

PRODUCT DEPENDENCIES

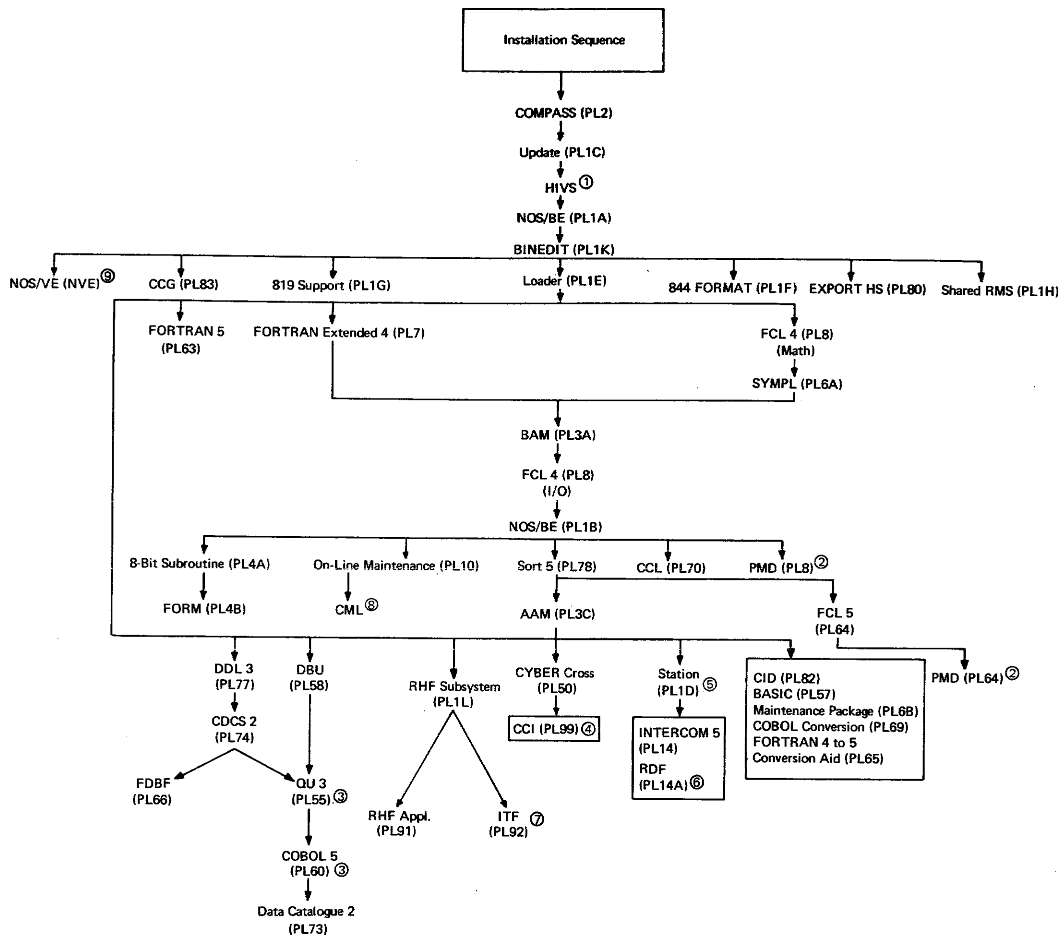
When the installation of one product requires the output tape from another product as its input, upgrade that tape to the level of the system being built. Some products are stacked on a single release tape, and therefore you must take care to ensure the integrity of the final output tape during the installation process. For example, you must assign the output tape from SYMPL as input to the maintenance package, and you must assign the output tape from 8-bit subroutines as input to FORM. For those sites with available disk space, the Update directive =DF SAVE creates intermediate permanent files to automatically satisfy these dependencies.

Several types of product dependencies determine the installation order of the NOS/BE system. These dependency types are:

- Compilers, assemblers, and utilities required by subsequent products.
- Relocatable binaries required for subsequent absolute module formation.
- Assembled texts and loaders required by subsequent products.
- COMPASS XTEXT relationships.
- Relationships such as multiple oldpl source dependencies.

Table I-1-1 describes the first two dependency types. Figure I-1-1 shows the recommended installation sequence, considering all the previously listed dependency types.

The flow in the recommended installation sequence is optimized for simultaneously processing as many decks as possible for efficient machine use. Install products in sequence moving from the top to the bottom of figure I-1-1. You can simultaneously install products on the same horizontal level or within a box. Every product is depicted in figure I-1-1, even though many products are optional and some are mutually exclusive. Requirements for a product are briefly described in table I-1-1 and described in detail in part II of this document.



1. The HIVS job (deck HIVSE on the installation deck PL) catalogs CTIBIN and EDITLIBs CTITEXT. The HIVS tape has been changed and renamed the CYBER Initialization Package (CIP) tape. All non-180-class mainframe customers should follow the same procedure as before.

2. The PL8 and PL64 jobs produce functionally equivalent postprocessors, either of which will work for FORTRAN 4 or 5 jobs. Only one needs to be built/installed.

3. COBOL 5 can use the DDL 3 interface. This interface is required for COBOL 5 if the CDCS option is selected.

4. CCI is required for INTERCOM 5 installation.

5. INTERCOM depends on Station only if CIOCP is defined during the installation of INTERCOM 5.

6. PL14A should be installed only on 180-class mainframes that do not install PL14.

7. Although INTERCOM 5 is required for ITF operation and verification, it is not required for ITF installation.

8. CML applies to CDC licensed maintenance customers only.

9. NOS/VE (NVE) applies to CDC licensed NOS/BE - NOS/VE dual state customers only.

Figure I-1-1. Installation Sequence

Table I-1-1. Installation Requirements (Sheet 1 of 3)

Product to Be Installed	Required in the Running System or User Library																
	COMPASS	CRM Basic	CRM Advanced	FORTRAN 4 Compiler	FORTRAN 4 Library	SYMPL	CYBER Cross System	DBU	CCG	8-Bit	DDL 3	CCI	COBOL 5	CDCS 2	Sort/Merge 5	FORTRAN 5 Compiler	FORTRAN 5 Library
NOS/BE (PL1A)	I																
NOS/BE (PL1B)	I	I		I	I	I											
NOS/VE																	
Update (PL1C)	I																
CMM (PL1C)	I																
Enhanced Station (PL1D)	I		I	I1	I1	I											
CYBER Loader (PL1E)	I																
844 Format (PL1F)	I																
819 Subsystem (PL1G)	I																
Shared RMS (PL1H)	I																
Gemini (PL1H)	I																
BINEDIT (PL1K)	I																
RHF (PL1L)	I					I											
COMPASS	I																
CRM Basic (PL3A)	I					I											
CRM Advanced (PL3C)	I	B		I	I,O	I									B		
8-Bit Subroutines (PL4)	I	E			E												
FORM (PL4)	I	B	O		E	I				E							
On-Line Maintenance Software (PL5A)	I	I														I	I
HIVS (HIVSE)	I																
SYMPL (PL6)	I				I2	I											
Maintenance Package (PL6)	I	I	I	I	I												
FORTRAN Extended 4 Compiler (PL7 or PL71)	I																

Table I-1-1. Installation Requirements (Sheet 2 of 3)

Product to Be Installed	Required in the Running System or User Library																
	COMPASS	CRM Basic	CRM Advanced	FORTRAN 4 Compiler	FORTRAN 4 Library	SYMPL	CYBER Cross System	DBU	CCG	8-Bit	DDL 3	CCI	COBOL 5	CDCS 2	Sort/ Merge 5	FORTRAN 5 Compiler	FORTRAN 5 Library
FORTRAN Library 4 (PL8)																	
Math part	I																
Remaining FCL	I	B		I													
FORTRAN 4 Post Mortem Dump Utility (PL8A)	I	B		I	I												
Sort/Merge 5 (PL78)	I	I,O															
INTERCOM 5 (PL14)	I	I		I	I								E				
RDF (PL14A)	I					I											
CYBER Cross System (PL50)	I	I		I	I												
QU 3 (PL55)	I	B	E		I	I		I			I,O		I,O	B			
BASIC 3 (PL57)	I					I											
DBU (PL58)	I	B	E		I	B											
COBOL 5 (PL60)	I	B	O		O	I					I 3						
FORTRAN 5 Compiler (PL63)	I								I								B
FORTRAN 5 Library (PL64)	I	B															
FORTRAN 5 Post Mortem Dump Utility	I	B		I	I												
FORTRAN 4 to 5 Conversion Aid (PL65)	I	I	I	I	I												

Table I-1-1. Installation Requirements (Sheet 3 of 3)

Product to Be Installed	COMPASS	CRM Basic	CRM Advanced	FORTRAN 4 Compiler	FORTRAN 4 Library	SYMPL	CYBER Cross System	DBU	CCG	8-Bit	DDL 3	CCI	COBOL 5	CDCS 2	Sort/Merge 5	FORTRAN 5 Compiler	FORTRAN 5 Library
	Required in the Running System or User Library																
FDBF (PL66)	I	B	B		I	B					I				B		I
COBOL 4 to 5 Conversion Aid (PL69)	I	I	I	I	I												
CCL (PL70)	I																
Data Catalogue 2 (PL73)		B	E										B				
CDCS 2 (PL74)	I	B	B		I	B					I				B		
DDL 3 (PL77)	I	B				I											
EXPORT HS (PL80)	I																
CID (PL82)	I			I	B	I											
CCG (PL83)																	
RHF Applications (PL91)	I					I											
ITF (PL92)	I					I											
CCI (PL99)							I										

LEGEND: I = install, E = execute, B = install and execute, O = optional at execute time. All products require Update for installation and require NOS/BE for installation and execution. All product set installation jobs that run after the CYBER Loader is installed use the new loader for overlay and capsule generation.

NOTES:

1. Required for installation of enhanced station if ES4XREF is defined.
2. Math only.
3. COBOL 5 requires DDL 3 (selected at installation time) if the CDCS interface is selected.

INSTALLATION DECK NOMENCLATURE

The decks that correspond to products identified in figure I-1-1 and which are contained on the installation decks oldpl have the following name.

PLnujx

The characters in the deck name indicate the following.

Character	Description
PL	Prefix indicating installation or verification job.
n	PL number to which the job applies.
u	Optional unique character for the PL number.
j	Job type identifier; one of the following.

Identifier	Description
I	Installation job.
Im	Partial assembly or variant deck; m denotes an execution sequence number.
E	EDITLIB job.
O	Overlay reformation job.
T	Text redefinition job.
C	Catalog job.
V	Verification job.
x	Optional unique character for job type identifier.

The applicability of these jobs is described in the part II sections of this document that detail the installation procedures for the operating system and for each product set member.

The verification jobs contained on the installation deck PL are provided so that following completion of the installation processes and the generation of a deadstart tape, you can deadstart the new system and subject it to a validation exercise by selecting and executing these jobs. They provide a quick verification so that, at a minimum, the skeleton of the whole system is in place. They do not provide a comprehensive test.

AUTOMAT PROCEDURE

The AUTOMAT procedure makes it easier to extract installation jobs from the installation deck PL. It produces the selected deck on a local file. The following job catalogs the AUTOMAT procedure.

Job	Comments
AUTOM,T10.	
REQUEST(AAA,SN,*PF)	
COPYBR(INPUT,AAA)	
CATALOG,AAA,AUTOMAT,ID=INSTALL,XR=XYZ,PW=XYZ.	
7/8/9	
.PROC,AUTOMAT,DECK,TD=HY,OPTION1,OPTION2,...,OPTIONn.	
IFE,TD=HY,TRK7.	
UPDATE,Q,D.8,*==,I=DIREC7.	
ELSE,TRK7.	
UPDATE,Q,D,8,*==,I=DIREC9.	
ENDIF,TRK7.	
.DATA,DIREC7.	
=ID JOBCARD	
=D ACCOUNT.2	
DECK,TO,MT1.	
=DEFINE MT,OPTION1,OPTION2,...,OPTIONn.	
=DEFINE	Put invariant defines such as ULIB here
=COMPILE DECK	
.EOR	
.DATA,DIREC9	
=ID JOBCARD	
=D ACCOUNT.2	
DECK,TO,HD01.	
=DEFINE HD,OPTION1,OPTION2,...,OPTIONn.	
=DEFINE	Put invariant defines such as ULIB here
=COMPILE DECK	
6/7/8/9	

OBTAINING INSTALLATION JOBS

To process selected installation decks with appropriate options specified, run a job similar to:

```
AUTO,T100.
ATTACH,OLDPL,DECKS,ID=INSTALL.
REQUEST,COMPILE,*Q.
ATTACH,AUTOMAT,ID=INSTALL.
AUTOMAT,deck,density,option1,option2,...,optionn.
ROUTE(COMPILE,DC=IN)
6/7/8/9
```

Parameter	Description
deck	One of the decks available on the installation deck PL.
density	HY, HD, or PE tape density.
$option_i$	Any appropriate option other than tape density (described in Deck Options in this section.

This technique works for all installation jobs except DST1 and DST2, which you can manually manipulate to add 844 OSY, 844 OSZ, 885 OSJ, and/or 66X OMT controlware. In this case, either you can obtain selected jobs in punch card form from the installation deck PL with a job similar to the one following, or you can modify common deck CWARE to transfer this controlware from a local file. You can insert the ATTACH of the local controlware file at LOCALCC.1.

You can use the following job to punch any of the installation decks.

```
DECKS,T100.
ATTACH,OLDPL,DECKS,ID=INSTALL.
ATTACH,AUTOMAT,ID=INSTALL.
AUTOMAT,deck,density,option1,option2,...,optionn.
REWIND,COMPILE.
COPYBF,COMPILE,PUNCH.
6/7/8/9
```

Parameter	Description
deck	One of the decks available on the installation deck PL
density	HY, HD, or PE tape density.
$option_i$	Any appropriate deck option that is described in Deck Options in this section (not including tape density) and not defined in the AUTOMAT $procedure_1$.

The call to AUTOMAT changes for each job, but all other cards remain the same.

NOTE

Pass Update define directives only in the preceding AUTOMAT procedure. Change variables, such as Type 1, and Communications Control INTERCOM (CCI PL99), by replacing or adding cards directly to the appropriate job deck. For jobs that require many define directives, add more options with the following AUTOMAT control statement.

AUTOMAT,deck,density,$option$_1$,option$_2$,
...,option$_n$$.

If many options are required, use the following statement.

AUTOMAT,deck,density,
$option$_1$,option$_2$,...,option$_n$$.

DECK OPTIONS

Various options for which you can define values during extraction have been embedded in the job decks present on the installation deck PL under the control of =IF DEF directives. The listing from deck REASON, described previously, shows the decks affected by defining the following options.

Deck Option	Description
BASE	The default installation (as opposed to EDITLIB or verification) type jobs are constructed to attach and update a corrective code file for use as input to Update correction runs to produce upgraded product and system oldpls. Defining BASE during the extraction of installation jobs causes omission of the corrective code file steps. Refer to the appropriate software release bulletin for the products for which you should define BASE.
HY, HD, and PE	Selects the appropriate REQUEST or LABEL statements in the installation deck.

Option	Selects
HY	800 bpi, 7-track.
HD	800 cpi, 9-track.
PE	1600 cpi, phase-encoded, 9-track.

You must select one of these options.

LIST	Places output from FORTRAN and SYMPL compilations and COMPASS assemblies on file LIST.
OMT	Activates control statements to include the 66x (MTS) tape subsystem controlware in the file named CWARE from which all prefixed binaries are included. If you define OMT, the prefixed controlware is expected to exist on permanent file OMTCWARE, ID=CWARE for job DST1 or DST2. Otherwise, it is assumed that a local file named CWARE, which contains this prefixed binary, has been attached or created at LOCALCC.1. The default residency for 66x and 67x tape drivers is central memory.
OSJ	Causes the inclusion of 7155 controlware on the file named CWARE. If you do not define OSJ, the local file CWARE created at LOCALCC.1 must include prefixed OSJ controlware (refer to the OMT description). If you define OSJ, the prefixed binary is expected to exist on permanent file OSJCWARE, ID=CWARE.
OSY	Causes the inclusion of 7054 controlware on the file named CWARE. If you do not define OSY, the local file CWARE created at LOCALCC.1 must include prefixed OSY controlware (refer to the OMT description). If you define OSY, the prefixed binary is expected to exist on permanent file OSYCWARE, ID=CWARE.
OSZ	Causes the inclusion of 7154 controlware on the file named CWARE. If you do not define OSZ, the local file CWARE created at LOCALCC.1 must include the prefixed OSZ controlware (refer to the OMT description). If you define OSZ, the prefixed binary is expected to exist on permanent file OSZCWARE, ID=CWARE.

Deck Option	Description
ECS	Activates control statements to accommodate assembly of the CMR segments and creates library CMRLIB to hold the relocatable binary.
NAD170	Activates control statements to include NAD-170 controlware. If you define NAD170, the prefixed binary is expected to exist on permanent file NAD170CWARE, ID=CWARE for job DST1 or job DST2. If you do not define NAD170, NAD-170 controlware is neither added to nor replaced in NUCLEUS library.
NADIBM	Activates control statements to include NAD-IBM controlware. If you define NADIBM, the prefixed binary is expected to exist on permanent file NADIBMCWARE, ID=CWARE for job DST1 or job DST2. If you do not define NADIBM, NAD-IBM controlware is neither added to nor replaced in NUCLEUS library.
NADMIN	Activates control statements to include minicomputer controlware. If you define NADMIN, the prefixed binary is expected to exist on permanent file NADMINCWARE, ID=CWARE for job DST1 or job DST2. If you do not define NADMIN, minicomputer controlware is neither added to nor replaced in NUCLEUS library.
OPTFTN	Allows installation of the normal, two-pass, optimizing FORTRAN Extended 4 compiler. If you are installing only the time-sharing (TS) mode, do not define OPTFTN.
TSFTN	Permits installation of TS mode of the FORTRAN Extended 4 compiler. If you do not define TSFTN, FORTRAN Extended 4 is installed without the capability of TS mode. In addition to TSFTN, you must define OPTFTN for proper installation of any product written in the FORTRAN Extended 4 language.

NOTE

If you define neither TSFTN nor OPTFTN, the deck PL71 will not produce a compiler. When you define both TSFTN and OPTFTN, as is applicable to the time-sharing compiler only, a compiler capable of handling both modes is installed.

Deck Option	Description
63CSET	Activates control statements to accommodate 63-character set installations and nominal execution field lengths required to accommodate conversion tables from a 64-character set to a 63-character set.
DEBUG	Activates control statements for the generation of CDCS 2 flow points in PL74I. These flow points trace the execution of CDCS modules from initialization to termination. However, the generation of flow points increases the execution size of CDCS by approximately 2500 octal words.

Deck Option	Description
DMGMNT	Activates assembly options to include the CDCS interface in COBOL 5; also affects nominal execution field length values. Installation of QU 3 requires defining DMGMNT for DST1, DST2, and DST3. To activate the CDCS 2 processing in COBOL 5, you must define the value CD2 in conjunction with DMGMNT in PL60I and PL60I1.
LOCLIB	Activates control statements in PL60I, PL60E, PL60O, PL66I, and PL66E to EDITLIB the binaries of the product into a local library instead of DST1 user libraries or the running system. Defining LOCLIB in PL73I activates control statements to attach the COBOL 5 local library created by PL60I, PL60E, PL60O, or PL60I1 and to execute this version of COBOL 5 during the installation of Data Catalogue 2.
SRMS	Can be defined in conjunction with setting installation parameter IP.SRMS to 1 in deck IPARAMS on PL1A and running job PL1H1 prior to installing NOS/BE PL1B with SRMS defined. PL1B1 accesses SRMSTXT during the assembly of 1PC-CVL.
NOCCP	Activates selection of the 77K PASCAL compiler and 77K PASCAL cross-reference program (125K default).
CIOCP	Activates inclusion of the 7000 connected I/O MUJ HELLO7. Use of this symbol requires that the updated PL1D (created by PL1DI) be assigned as input to PL14I.
CATALOG	Causes COBOL 4 to 5 conversion aid permanent files to be cataloged by job PL69I; causes CYBER Cross System permanent files to be cataloged by job PL50I for subsequent installation of CCI3.
ULIB	Invokes the use of the user library method for deadstart tape creation.
ES4IMS	Activates control statements to produce IMS documentation for the enhanced station.
ES4XREF	Creates a global cross-reference for the enhanced station SYMPL routines (MFSTAT and SPOT jobs). Defining this symbol requires that Sort/Merge 5, FORTRAN Extended 4, and FORTRAN Common Library 4 be present in the running system.
DIM	Activates the capture of on-line maintenance software deadstart records.
CMU	Activates inclusion of CMS test for compare/move unit hardware.
SAVE	Reduces the number of tape assignments. All I decks that produce a tape used by another I deck catalog the necessary information as a permanent file. The receiving I job attaches this file instead of requesting the tape. Thus, all I jobs require one input tape and one output tape. In DST1, all user libraries are purged unless SAVE is defined. In DST2, the only tape assignment required is the output deadstart tape.
NOINT	Avoids attempting to install 2550 NPU driver and multi-user jobs. You must define it when installing PL14A.
CTI	Activates control statements to include CTI on deadstart tape. This option should only be defined for sites receiving a CIP700 tape.

COMMON DECK MODIFICATIONS

ACCOUNT — Allows insertion of a job statement and a valid ACCOUNT statement (=D ACCOUNT.2); a common deck is called by all job decks.

EXIT — Allows use of the console display for abnormal termination of installation decks. You can modify it to provide different termination procedures. The following control statement sequence is suggested when reporting possible installation problems.

```
=I EXIT.1
REWIND(INPUT)
COPYSBF.
EXIT(S)
REWIND(INPUT)
COPYSBF.
```

LOCALCC
LOCALIN
READS — Allow local source code modifications. Every I-suffixed deck contains calls to these three common decks. Insert additional control statements LOCALCC.1, insert additional input sections (each section preceded with an =WEOR,0 UPDATE directive) at LOCALIN.1, and insert any UPDATE *READ directives at READS.1. The following examples illustrate the use of this facility.

Local modifications on an UPDATE oldpl:

```
=I LOCALCC.1
ATTACH(OLDPL,LOCPL,ID=LOCID)
UPDATE(Q,C=LOCPL1A)
RETURN(OLDPL)
=I LOCALIN.1
=/ NEED =WEOR, 0 TO MAINTAIN INPUT FILE POSITION
=WEOR,0
*C LOCPL1A
=I READS.1
*READ LOCPL1A
```

Local modifications in source format:

1. ```
   =I LOCALCC.1
   ATTACH(LOCPL1A,ID=PL1A)
   =I READS.1
   *READ LOCPL1A
   ```

2. ```
   =I LOCALCC.1
   ATTACH (LOCCODE,LOCPL1A, ID=PL1A)
   ```
   (The default file name LOCCODE is always read.)

3. ```
   =I READS.1
   *IDENT LOCPL1A
   ```
   .
   .   (Cards in extraction deck or procedure.)
   .

SAVEPL — Allows you, through the use of an =I SAVEPL.2 directive, to itemize or add additional operations to a new program library tape before it is returned. The newpl always has an lfn of PLxx at this point in the program.

Example:
```
=ISAVEPL.2
ITEMIZE,PL1A,E,N.
```
```

DST1 AND DST2 COMMON DECKS

Common Deck Name	Use
CWARE (CWARE.2)	Common deck location containing a TRANSFER(*,CWARE) directive for inclusion of prefixed controlware on the new deadstart tape. Refer to OMT, OSJ, OSY, and OSZ under Deck Options.
CMRS (CMRS.2)	Common deck for inserting TRANSFER directives to include additional CMRs (=I CMRS.2).
XTRALIBS (XTRALIBS.1)	Empty common deck for adding additional system libraries. (A LIBRARY, REPLACE or ADD, and FINISH span is required for DST1; an INCLUDE is required for DST2.)
MORENUC (MORENUC.1)	Empty common deck for adding additional NUCLEUS routines (with REPLACE directives).
MOREOV (MOREOV.1)	Empty common deck for adding additional overlays to SYSOVL (with REPLACE directives).
MORESYS (MORESYS.1)	Empty common deck to add more SYSLIB routines (with REPLACE directives).
MOREPP (MOREPP.1)	Empty common deck to add more PP routines (with REPLACE directives).
LOCALCC (LOCALCC.1)	Empty common deck for insertion of control statements to access additional files.
SYSPROC (SYSPROC.2)	Empty procedure (SYSPROC) to which the user may add his/her own procedure control statements at =I SYSPROC.2. This procedure will be called after each deadstart.

SPECIAL PURPOSE DECKS

MINI

Use this deck to save the operating system and product set corrective code files from the BCC tape as permanent files. The following files are cataloged.

BCC Tape File Number	File Name	Description
1	OSMINIT	Operating system code
2	MINIT	Product set code
8	RHFMNIT	RHF code

You must run MINI before all the installation decks that require corrective code, since those decks attempt to attach OSMINIT or MINIT.

ULIB

Run this deck prior to the deadstart tape creation job (DST1), if you are using the user library approach to installation (selected by including the =DF,ULIB directive when obtaining the installation jobs). This deck manipulates certain user library files in preparation for DST1. It combines files USERCC and USERNUC and creates a second cycle of files USERPP and USERPS. The second cycles of USERPP and USERPS are sequential files.

NOTE

Take a permanent file dump of the user libraries (ID=CCT) prior to running ULIB. If DST1 fails or if another product is to be installed, you can reload the permanent files and run ULIB and DST1 again.

LIBS

Run this deck first if you are using the user library approach to installation. This deck catalogs all possible user libraries and inserts a dummy routine (ZZZ) in each. This dummy routine is later deleted from the libraries.

DST1

This deck is a deadstart tape creation job which generates a system from user libraries. This deck is applicable only if you are using the ULIB approach (=DF,ULIB directive included) for the other installation jobs.

DST2

This deck has two purposes, depending on whether you are using the running system method or the ULIB method of installation. If you are using the running system method of installation, this deck creates a deadstart tape from the running system replacing the existing PL1A, PL1B, PL1E, and PL5 binaries with the updated versions. If you are using the ULIB method of installation, this deck copies the operating system binaries, created in a previous assembly, into the appropriate user libraries. This eliminates the necessity of reassembling the operating system in order to install subsequent products. No deadstart tape is created using the ULIB approach.

NOTE

Jobs DST1 and DST2 contain EDITLIB comment directives indicating alternative choices regarding central memory residency and INTERCOM driver choices. Users of these job decks are encouraged to review these comments before running the job decks.

DST3

This deck creates a deadstart tape from the running system, and Control Data optionally replaces the deadstart diagnostic sequencer routines using existing PL5 binaries. Model 171 sites with the no card reader configuration use the AUTO procedure to initiate this deck; when initiated, a CMR configured for a 2550 multiplexer is assembled and placed as CMR zero on the new deadstart tape. All other site configurations are discouraged from using this installation option.

IPTEXT

This deck is a utility that replaces the running system text of the same name. Running IPTEXT ensures that references to any new or altered IPARAMS are accommodated during the assembly of the system and products (refer to 63CSET). Whenever IPARAMS IP.C63, MODEL, OS.NAME, or OS.VER is to be changed from its default value when PL1AI is run, you should run IPTEXT. If you do not run IPTEXT at this time, the altered values are not reflected in UPDATE and COMPASS.

COMPCOM

This deck creates and catalogs the COMPCOM file. Because this file is produced in the PL2I job, this deck need only be used if PL2I is not run or if the file is destroyed. The file COMPCOM is used by the jobs that create the FORTRAN Extended 4 compiler (PL7).

PL1AT

Job PL1AT is a utility deck used to equate SYSTEXT to CPCTEXT. None of the installation decks reference SYSTEXT; therefore, this job is not needed for the installation process. The default equivalence of SYSTEXT is IOTEXT, as installed by CYBER Record Manager installation decks and present on the unconfigured deadstart tape. You can run this job after the installation of CYBER Record Manager and before the creation of a deadstart tape.

SYSTEM TEXTS

Common decks included on the NOS/BE and CYBER Record Manager program libraries are combined to form 15 system texts. The source location and contents of these common decks are:

Deck	Location	Contents
ACTCOM	PL1A program library	CPU program system action request macros
COMACIO	PL1A program library	CPU input/output macros
COMAFET	PL1A program library	File environment table generation macros
COMAREG	PL1A program library	Replacement for R=pseudo instruction
COMCECM	PL1B program library	Macros to redefine RE and WE instructions for interpretive ECS access
COMSHSP	PL1A program library	819 RMS definitions and macros
COMSRAS	PL1A program library	System communication symbols
CPSYS	PL1A program library	CPU input/output macros using CPC
IPARAMS	PL1A program library	NOS/BE 1 installation parameters
LMACOM	PL1A program library	CPU program loader request macros
PFCOM	PL1B program library	Permanent file macros
PPSYS	PL1A program library	PPU system definitions
SCHCOM	PL1A program library	Integrated scheduler macros
SISICOM	PL1A program library	Indexed sequential macros
STATCOM	PL1A program library	Enhanced station symbol definitions
CRMCOM	CYBER Record Manager program library	CYBER Record Manager user macros

Table I-1-2 shows the combination of these common decks into the various system texts required for full use of the product set. These texts are fixed in content except SYSTEXT; as released, SYSTEXT contains ACTCOM, COMSRAS, and CRMCOM. At your option, SYSTEXT may contain COMAFET, CPSYS, and SISICOM in lieu of CRMCOM.

Table I-1-2. Common Decks and System Texts

Common Deck	System Text Name													
	CMRTEXT	CPCTEXT	CPUTEXT	IOTEXT	IPTEXT	LDRTEXT	PFMTEXT	PPTEXT	SCHTEXT	SCPTEXT	STATEXT	SYSTEM DEFAULT	SYSTEM ALTERNATE	ECSTEXT
ACTCOM		X	X	X						X		X	X	
COMACIO			X											
COMAFET		X	X							X			X	
COMAREG			X											
COMCECM														X
COMSHSP	X													
COMSRAS		X	X	X				X		X		X	X	
CPSYS		X								X			X	
IPARAMS	X				X									
LMACOM						X								
PFCOM							X							
PPSYS								X		X				
SCHCOM									X					
SISICOM		X											X	
STATCOM											X			
CRMCOM				X								X		

The following text is installed when you install CYBER Record Manager, and FORTRAN Extended 4.

FTNMAC	COMPASS routines generated by the FORTRAN Extended 4 compiler in E mode require the specification of this text when assembled.

The following text is cataloged when Shared RMS (PL1H) is installed.

SRMSTXT	System routine MNT on PL1B requires the specification of this text when assembled if IP.SRMS is set to 1. The INTERCOM 5 routine MYQ on PL14 requires this text when assembled if Gemini is to be used on the system. The routine Gemini on PL1H also requires this text.

The system texts are constructed as a part of the installation process.

Table I-1-3 shows the product name, the program library number, the section in Part II in which the product is discussed, the BCC deck identifier, and the definable attribute causing the inclusion of the product in the DST jobs.

Table I-1-3. Product Installation (Sheet 1 of 3)

CDC Product	PL	Section	BCC Deck Identifier	DST Define†
BASIC 3	PL57	21	BAS	BAS
BINEDIT	PL1K	1	PL1K	-
COBOL 4 to 5 Conversion Aid	PL69	25	C45	
COBOL 5	PL60	24	CL5	CL5
Common Code Generator	PL83	1	CCG	
Communication Control INTERCOM 3	PL99	29	CC3,SCF	CCI2550
COMPASS 3	PL2	2	CPS	-
CYBER Control Language	PL70	26	CCL	-
CYBER Cross System 1	PL50	15	XSY	

†A dash in this column means that the product is automatically included or is not applicable; a blank in the column means that the product is only cataloged on the system by the installation catalog job.

Table I-1-3. Product Installation (Sheet 2 of 3)

CDC Product	PL	Section	BCC Deck Identifier	DST Define†
CYBER Database Control System 2	PL74	30	CD2	CD2
CYBER Interactive Debug 1	PL82	32	ID1	ID1
CYBER Loader 1	PL1E	16	LDR	-
CYBER Record Manager 1 Basic	PL3A	3	SW1	-
CYBER Record Manager 2 Advanced	PL3C	4	AM2	AM2
Data Base Utilities 1	PL58	22	DBU	DU1
Data Catalogue 2 Version 1	PL73	41	DC2	-
Data Description Language 3	PL77	31	DL3	DL3
Enhanced Station	PL1D††	1	ES4	ES4
EXPORT High Speed	PL80	40	EHS	EHS
Factory Format Support (844-21 and 844-4x)	PL1F	17	FMT	-
FORM 1	PL4B	6	FO4	FO4
FORTRAN Database Facility 1	PL66	35	FDB	FDB
FORTRAN Extended 4 (compiler)	PL7	9	FCC	
FORTRAN Extended 4 (time-sharing option)†††	PL7A	9	FCC	
FORTRAN 4 Common Library	PL8	34	FCL	-
FORTRAN 4 Post Mortem Dump Utility	PL8	34	PMD	-
FORTRAN 4 to 5 Conversion Aid	PL65	37	F45	F45
FORTRAN 5 (compiler)	PL63	38	FC5	FC5
FORTRAN 5 Common Library	PL64	39	FL5	FL5

†A dash in this column means that the product is automatically included or is not applicable; a blank in the column means that the product is only cataloged on the system by the installation catalog job.
††PL1D and PL1H compose the Multimainframe Module 1 package.
†††TSFTN governs the default field length of FTN. OPTFTN causes the inclusion of FTNMAC (refer to section II-9).

Table I-1-3. Product Installation (Sheet 3 of 3)

CDC Product	PL	Section	BCC Deck Identifier	DST Define†
FORTRAN 5 Post Mortem Dump Utility	PL64	39	PMD	-
Gemini	PL1H	1	SC4H	-
Interactive Transfer Facility of RHF	PL92	43	ITB	ITF
INTERCOM 5	PL14	28	IN5	IN5
Maintenance Package (including SYMPL)	PL6	8	CA4,SMP	CA4
NOS/BE (part A)	PL1A	1	OSA	-
NOS/BE (part B)	PL1B	1	OSB	-
On-Line Maintenance Software (CE diagnostics)	PL5A	7	DIM	DIM
Query Update 3	PL55	19	QU3	QU3
Remote Host Facility Subsystem	PL1L	42	RHF,RHC	RHF
Remote Host Facility Applications	PL91	42	RHP	RHF
Shared RMS (844-21 and 844-4x)	PL1H††	1	SC4H	-
Sort/Merge 5	PL78	11	ST5	ST5
Update 1, CYBER Utilities 1, Common Memory Manager 1	PL1C	1	UPD	-
8-Bit Subroutines 1	PL4A	5	BE4	BE4

†A dash in this column means that the product is automatically included or is not applicable; a blank in the column means that the product is only cataloged on the system by the installation catalog job.

††PL1D and PL1H compose the Multimainframe Module 1 package.

PART II

NOS/BE VERSION 1.5 1

HARDWARE CONFIGURATION

The target minimum hardware configuration consists of:

- Either one CYBER 180 Computer System, one CYBER 170 Computer System, one CYBER 70 Computer System, or one 6000 Computer System with a minimum of 65K central memory and 10 peripheral processor units.
- Either one 844 disk subsystem (844-21 or 844-4x) or one 885 disk subsystem.
- Either one 580 printer or one line printer on a 734-1/CDC CYBER 18 terminal that is physically located with the central computer and is driven by a 255x communication subsystem.
- Either one 405 card reader or a card reader on a 734-1/CDC CYBER 18 terminal that is physically located with the central computer and is driven by a 255x communication subsystem. (The terminal card reader can be used only for source deck submission and not for binary decks.)
- Two magnetic tapes from either 667, 669, 677, or 679.
- One 255x communication subsystem with at least 65K memory (required if the site does not have a 405 card reader or if the site requires INTERCOM 5).

RELEASE MATERIALS

Materials in the NOS/BE 1 release package consist of the following.

PL1A, PL1B	NOS/BE program libraries
PL1C	Update release tape, including COPYL, ITEMIZE, and CMM
PL1D	Station release tape†
PL1E	CYBER Loader 1 release tape
PL1F	Factory Format Support (844-21 and 844-41)
PL1H	Shared RMS release tape†
PL1K	BINEDIT release tape
PL2	COMPASS 3 release tape
PL3A	CYBER Record Manager Basic Access Method 1 release tape

†Separately licensed products, although PL1D and PL1H together compose the Multimainframe Module 1 package.

PL3C	CYBER Record Manager Advanced Access Method 2 release tape
PL4	8-Bit Subroutines 1 and FORM 1 release tape
PL5A	On-Line Maintenance Software release tape
PL14A	Remote Diagnostic Facility (180-class, 865 and 875 mainframes only)
PL70	CYBER Control Language release tape
	Unconfigured deadstart tape
	Installation deck program library
	Small binary coldstart card deck
	CIP install tape
	Binary patch tape

For model 176 installations, the following is also included in the operating system release package.

PL1G	819 RMS

PL1G is a three-file tape, structured as follows:

File	Description
1	Program library.
2	Binary of the 819 FLPP driver.
3	Binary of CMR segments to support 819 rotating mass storage (RMS).

For 180-class, 865 and 875 mainframes, the following is also included in the operating system release package.

PL14A	Remote Diagnostic Facility (RDF) drivers. If INTERCOM 5 (PL14) is installed, PL14A is not needed since RDF is included on PL14.

The NOS/BE program library tapes (PL1A through PL1K) contain the source programs for all routines composing NOS/BE 1. PL1A and PL1B contain one file each; assembled binary is not included on these tapes. PL1C contains a program library as file 1, the absolute binaries of UPDATE, COPYL, and ITEMIZE as file 2, and the relocatable binaries of CMM as file 3. PL1E, PL1F, and PL1H contain a program library as file 1 and assembled binary as file 2. PL1D is a six-file tape structured as follows: file 1 is a program library; files 2 through 6 contain PPU absolute binaries, station absolute binaries, spun-off task absolute binaries, station relocatable binaries, and spun-off task relocatable binaries, respectively. PL1K contains the on-line patch capability for the CTI binary. The CYBER Initializaton Package (CIP) tape contains the HIVS and CTI routines on file 1 and the CTITEXT on file 2.

The unconfigured deadstart tape contains only the products NOS/BE (including Update, CMM, and CYBER Loader, but excluding the station and shared RMS), COMPASS, BINEDIT, Factory Format Support, On-Line Maintenance Software, CYBER Record Manager Basic Access Method, SYMPL, and CYBER Control Language. The unconfigured deadstart tape corresponds to the release program libraries. Three CMRs are present on this tape. The first reflects a blank EST, and the second and third reflect released configurations with model 176 capabilities. The first CMR allows keyboard entries of up to eight tape drives and six RMS equipments. While the released CMRs are generally usable for channel and equipment numbers, these CMRs allow up to 8 tape drives and 12 RMS equipments.

The small binary coldstart card deck is for 66x coldstart-type deadstarts. If it is used, the 66x unprefixed controlware deck must follow. (A tape coldstart using the 7152 Mass Storage/Magnetic Tape Controller is described in the NOS/BE Operator's Guide.)

If deadstart diagnostics are to be available or if you are deadstarting a 180-class mainframe, use the CIP tape to load the HIVS module onto disk. Refer to the CYBER Initialization Package (CIP) User's Handbook for further information.

If you are deadstarting your mainframe with the unconfigured deadstart tape, you must use the CIP tape to load the CTI module onto disk, since the unconfigured deadstart tape does not contain CTI. If your machine is not a 180-class mainframe, you can specify =DEFINE CTI when building a system to generate a deadstart tape with CTI.

Content, structure, and use of the COMPASS, CYBER Record Manager, FORM, and CYBER Control Language release tapes are discussed in the section devoted to each of these products.

Required supplements to this package include the following.

PL6	SYMPL 1 Maintenance Tools program library
PL7	FORTRAN 4 compiler program library
PL8	FORTRAN 4 object routines program library
PL63	FORTRAN 5 compiler program library
PL64	FORTRAN 5 mathematical and I/O routines program library

Provided as a required supplement to the basic release package, PL6 (SYMPL) is needed for use in installing the NOS/BE system (PL1B), the enhanced station, FORM, SYMPL, BAM, AAM 2, CDCS 1, DDL 3, DBU, COBOL 5, BASIC, CDCS 2, and QU 3. PL7 and PL8 are necessary for complete installation of PL1B and numerous other products in the total product set. PL63 and PL64 are necessary for complete installation of PL5A.

CIP INSTALL TAPE

CIP provides deadstart interface routines (CTI (Common Text and Initialization), EDD (Express Deadstart Dump), and so forth), disk and tape conrolware, and text (CTITEXT) that contains symbols necessary for building NOS/BE routines (on PL1A) which communicate with CTI during deadstart. Refer to the CYBER Initialization Package User's Handbook for further information about the structure and use of the CIP install tape.

Use the GETCW procedure on the CIP tape to move controlware from the CIP tape to NOS/BE permanent files for use by jobs DST1 and DST2. Further details about this procedure are found in the CIP release materials.

Run job HIVS, from the installation deck PL, before job PL1AI. If CTI is defined, HIVS catalogs a permanent file, CTIBIN, ID=CCT, for input to job DST1. The job also EDITLIBs CTITEXT into either USERNUC or NUCLEUS, depending on whether or not the user library method is being used. CTITEXT is then used by job PL1AI.

NOTES AND CAUTIONS

The CEJ/MEJ switch must be enabled for all machines.

Gap sectors on 844 devices were eliminated at NOS/BE 1.4, level 508. Therefore, if an 844 device is initialized on a level 508 or later system, the user cannot read or write this device at a previous system release level. However, at level 508 the user can read and write 844 devices with gap sectors which are created (labels written) at a previous system release. In other words, a user can upgrade to level 508 or later without reinitializing 844 devices.

With the elimination of gap sectors, the PB size of an 844 device, as seen by a user, changes from 112 to 114 PRUs, and the choices of nonstandard RB sizes which divide the PB evenly are restricted to 6, 19, and 38. However, the user can still choose from RB sizes of 4, 7, 8, 14, 16, and 28 (which divide 112 evenly), leaving the total space available on the device the same as it was at previous releases and not increased by the elimination of gap sectors.

NOS/BE is released with all software assembled to support integer multiply. To ensure the proper execution of all code related to this hardware capability, install the following change orders.

Computer Models	Change Order Number
6600, models A, B, C	CA26938, CA30886
6600, models D, E	CA26379, CA31029
6200/6400/6500, all models	CA26792, CA30638, CA28539
6700, models A, B, C	CA27065, CA30966
CYBER 70, models 72 and 73	CA30639
CYBER 70, model 74	CA31029

In addition, install the following FCOs listed. These FCOs will make underflow results obtained from normalized numbers with zero exponents positive to ensure consistency across the product lines. The lack of these FCOs will not have a negative impact on the operating system or product set members; however, the CE diagnostic routine, CT3, may fail if a certain set of random operands are generated for the multiply unit. For additional information concerning CT3 and integer multiply, consult the discussion of installation parameter INTIMULT in the NOS/BE On-Line Maintenance Software Reference Manual.

Computer Models	Change Order Number
6200/6400/6500, all models	CA33439
6700, models A, B, C	CA33439
6700, model D	CA32988
CYBER 70, models 72 and 73	CA32988

The deadstart tape and the system device must be on different channels. This is necessary because deadstart keeps two data streams going at one time.

When NOS/BE is run on a 6500 or 6700 using IP.XJ=1 or 2, install FCO CA23065 to prevent both CPUs from being in monitor mode simultaneously.

Extended memory I/O buffering and swapping of non-MUJ jobs to extended memory cannot be activated if 819 RMS is installed. L.ECSSWP defaults to 20B when 819 RMS is installed to allow MUJ jobs to swap to extended memory.

Extended memory I/O buffering is not supported for private device sets.

If the OUTPUT file is rewound but no other action is performed on the file, the OUTPUT file will be evicted; a skip to EOI is not performed prior to writing the job dayfile on OUTPUT.

You must observe the values BASE and IRADR as described under Deadstart Installation Parameters in this section.

Support of the 6603, 6638, 821, and 854 devices is not provided.

Support of the 7611-11 Station is not provided.

The NOS/BE 1.5 release contains a CYBER Initialization Package (CIP) on a separate tape for each mainframe. You cannot install the CIP package on a deadstart tape for 180-class mainframes; you must install on disk to be executed, even if deadstart from RMS is not to be used.

Device overflow is not allowed between device sets.

A private device set may contain 841 devices, 844-2x devices, 844-4x devices, or 885 devices, but each private set may contain only one type of device.

On a model 74-28 install FCO CA35742 to avoid a CPU A hang with monitor abort, which will otherwise occur when a user job executing in CPU A under control of MODE,0. executes an instruction that attempts an out-of-range address.

Install FCO CA36100 on the 844 Buffer Controller (FA710 or FA719) to use the A08 controlware.

Code is activated to use FCO CA37722 (PP halt on CM read error), applicable to model 176 and any other CYBER 170 model at production level C or D. Although this FCO is not essential to system operation, it improves system reliability and you should, therefore, install it.

When running on a mainframe with 4xPPs and a 7154 Disk Controller configuration, install the 7154 special option 65258-1. This option resolves 4XPP speed timing problems associated with data parity checking.

NOTES AND CAUTIONS (SHARED RMS ENVIRONMENT)

The available RB count displayed in the DSD V display for shared devices is not assured to be accurate at all times.

A public device set with the system set attribute cannot be shared.

FCO CA35682, FCO CA35683, and the latest release of the 844 controlware are required for shared RMS.

If RMS deadstart is used, it is recommended that neither the RMS deadstart device nor any controller accessing the device be accessible from another mainframe.

NOTES AND CAUTIONS (TAPE SCHEDULING)

The installation parameter IP.SCHDE determines whether 9-track units are scheduled by device or by density.

Setting IP.SCHDE to a nonzero value enables automatic scheduling by density. The job statement parameters for 9-track tape resources can be NTk, HDk, PEk, or GEk, where k is the number of tape units at each density to schedule for the job, and NT is equated to the installation default density as defined by parameter IP.NDEN in deck CIOCOM. When IP.SCHDE is nonzero, jobs containing request for 9-track tapes with the density specified as other than the default density must include the corresponding density specifications on the job statement.

Setting IP.SCHDE to zero disables automatic scheduling by density. Tape job statement parameters NT, HD, PE, or GE are allowed but are not required. Nine-track tape units scheduled as HD, PE, or GE are added to the count of units scheduled as NT.

NOTES AND CAUTIONS (ATS 679 TAPE DRIVES)

If tape resource scheduling by density is not enabled (IP.SCHDE=0), deadlocks (which may require jobs to be rerun or dropped) may occur because 679/GCR (group coded recording) tape drives do not have 800-cpi recording capacity, but are considered available for assignment to any 9-track tape request. Avoid deadlocks by one of the following operational procedures.

- Set the parameter IP.SCHDE to a nonzero value. Refer to Notes and Cautions (Tape Scheduling).

- Use tape previewing to manually schedule jobs requiring 679 tape drives. This requires that you adopt a VSN convention for 679 tape reels, enabling the operator to recognize requests for such tapes on the P display. You can manually initiate jobs after all tape resources have been made available. The limitation of this method is that all tape requests are not shown on the previewing display in the case of multimainframe systems, macro requests, or CCL procedure files.

- 679 scheduling deadlocks can also be avoided by ensuring that all 679 tape drives remain logically off in the equipment status table. In response to a request for a 679 tape, the operator must turn the tape drive logically on, assign it to the requesting job (or allow automatic assignment if labeled), and turn the drive logically off. Tape drive overcommitment scheduling cannot be used with this method (bit S.OCJI of IP.TSG must be set to 0).

Use of 6250-cpi density on 679-7 tape drives is not supported on CYBER 70 or 6000 mainframes. Use of 6250-cpi density in an unsupported configuration may result in lost data during deadstart.

Certify tape reels at 3200 fci or greater prior to recording at 6250-cpi.

INSTALLATION PROCEDURES

Installation of NOS/BE requires that you customize to conform to the site's hardware and software specifications by selecting:

- General installation parameters within IPARAMS.
- Tape processing installation parameters within CIOCOM.
- Miscellaneous installation variables.
- CMR configuration parameters.
- Deadstart installation parameters.
- Permanent file and device set installation parameters.
- Scheduling parameters.
- Extended memory installation parameters.
- Update and common memory installation parameters.
- Symmetric/Replacement Station installation parameters.
- CYBER Utilities installation parameters.
- Gemini load-leveling installation parameters

Once parameters have been selected, create configured program libraries and a deadstart tape by running the model jobs.

GENERAL INSTALLATION PARAMETERS (IPARAMS)

General installation parameters related to NOS/BE are defined within the COMDECK IPARAMS. IPARAMS is listed in the routines IPTEXT and CMR. Other installation parameters are described elsewhere in this and other sections of this document.

Assigned (default) values and descriptions follow. The parenthetical value is the default value as set on the released program library.

The default values of the IPARAMS configuration parameters are defined with the CEQU or CMICRO macros, so that an installation can insert all modifications at one given place. The CEQU and CMICRO macros are used to define variables conditionally. Since they are effective only if the variables have not been previously defined, any modifications must precede the default definitions.

Symbols can be defined by EQU or CEQU except for OSID, MODEL, and HF.LIST, which are micros and must be defined by CMICRO.

Examples of changes for deck IPARAMS:

```
*I    IPARAMS.15
IP.LINK     CEQU          0
HF.LIST     CMICRO        10,(P176,S10,L)
IP.CSET     EQU           IP.C64.2
```

The following list constitutes the extent of installation changeable symbols in IPARAMS. Changes to the default values listed should be made at IPARAMS.15 in an update of PL1A. The IPARAMS common deck also contains symbols IP.ILCMD, IP.IUSID, IP.1M1, IP.1WB, and IP.1ZZ. These symbols are described in the INTERCOM Version 4 section.

HF.LIST (P74,S7)

Micro whose value specifies the presence of certain hardware features in the configuration on which you are installing the product set. Always supply HF.LIST in addition to the MODEL micro, since use of various hardware features by the product set is conditional on HF.LIST. However, if you do not define HF.LIST, a default value that is based on the MODEL micro and assumes no optional hardware is used. The default HF.LIST based on MODEL is a temporary capability that will be removed in a future release. You can define the following entries in HF.LIST.

Entry	Description
C	Compare/move unit (CMU) hardware is present.
L	Extended memory is present which can be accessed by direct access instructions (014 and 015). This memory is large central memory (LCM) on model 176, and unified extended memory (UEM) on 180-class, 865 and 875 mainframes.

Entry	Description
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. Following are instruction stack loop sizes for the given mainframes.

Seven Words	Ten Words	Sixty-Four Words
6600	Model 175	Model 990
Model 74	Model 176	
	Model 740	
	Model 750	
	Model 760	

Px — Type of central processor; x can be one of the following values.

x	Description
S	Serial type CPU. Use S for 6200, 6400, 6500, models 71, 72, 73, 171, 172, 173, 174, 720, 730, 810, 815, 825, 830, 835, 840, 845, 850, 855, and 860.
74	6600, 6700, and model 74.
175	Model 175.
176	Model 176.
740	Model 740.
750	Model 750.
760	Model 760.
865	CYBER 170 model 865.
875	CYBER 170 model 875.
990	CYBER 180 model 990.

The processor type defaults to PS if you define HF.LIST but omit the processor type.

Entry	Description
PSD	Central processor's exchange package contains a PSD register. This exists on model 176 only.
CRW	Central memory read/write operations are performed for 660/670 instructions. This occurs only on 180-class, 865 and 875 mainframes.

Default values for HF.LIST are as follows:

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,PSD
720	C,PS
730	C,PS
740	P740,S10
750	P750,S10
760	P760,S10
810-860	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.

You can use a central processor type of PS, P74, or P175 when defining HF.LIST for a product set intended to be run on multiple mainframes. You can include stack size (even if not all the mainframes have a stack), but do not include C and L unless the respective features exist on all the mainframes in the configuration. The resulting product set will not necessarily perform optimally on any of the mainframes, but will perform better on a parallel processor (such as a 175) if that processor type is set in HF.LIST.

IP.CMU (0)

If nonzero, Compare/Move Unit hardware is present. If nonzero, the system will not run on a non-CMU mainframe (such as a model 175 or a 6600).

IP.ACNT (0)

If zero, normal control statement processing occurs. If set to 1, the job statement is copied to RA+70_8 through RA+77_8; the CPU program ACCOUNT is then loaded and executed. If set to greater than 1, the first statement following the job statement is copied to RA+70_8 through RA+77_8 before the CPU program ACCOUNT is loaded and executed. No dayfile message is issued when ACCOUNT is called.

If IP.ACNT=1 and IP.ARCH=1, follow the instructions in the following three paragraphs.

If an installation has system modifications which require accounting information before a job can come to a control point, change routine 1PF to insert appropriate information into the control stream of the job that performs archive file retrieval.

Insert any accounting information needed by the installation into the control statement buffer. Format these statements, including the job statement, according to installation procedures, using DIS or DATA statements. Each DIS or DATA that completes a card must be followed by a call to the PAD macro, which pads the card with zeros.

Set JOBNAME equal to a valid five-character, local file name to be used in setting up the input FNT for the archive retrieval job. 1PF adds two random digits to this jobname before storing it into the FNT. JOBNAME should be the same as that used on the job statement. No CM, tape, or priority requirements need be on the job statement, as 1PF sets up the input FNT with all such requirements satisfied.

Example 1 (add accounting information on job statement):

```
*D          1PF.246
CARD1       DATA    H*JBNME,CMTIME,*
*D          1PF.250
TAPEJ       DIS     ,*MT1.       ACCOUNTING INFORMATION*
```

Example 2 (add accounting information on account statement and change job name):

```
*D          1PF.246
CARD1       DATA    H*LODME,CMTIME,*
*D          1PF.250
TAPEJ       DIS     ,*MT1,*
            PAD
CARD2       DIS     ,*ACCOUNT(X,Y)*

*D          1PF.276
JOBNAME     DIS     ,*LODME*
```

IP.CP (6)

If set to 6, default punch mode is 026. If set to 9, default punch mode is 029. The alternative punch mode is selectable by ROUTE parameters.

IP.ARCH (1)

Archive feature; permanent files dumped under a mode 2 permanent file DUMPF

Entry	Description
0	No longer will have a PFC entry and will not be retrieved from tape at ATTACH time.
1	PFC entries are retained and are retrieved from tape at ATTACH time (refer to IP.ACNT).
2	PFC entries are retained but are not retrieved from tape at ATTACH time.

IP.CERNT (0)

Maximum number of messages (1 to 4095) that can be entered into the CERFILE by a single job. Only messages sent through ELM are counted. If zero, there is no maximum limit. The default value is equal to IP.MSCT.

IP.CPLM (5)

Installation-defined number of central processor (CP) seconds by which a job is incremented if the CP time limit (specified on the job statement of IP.STL) is exceeded or if the job requires EXIT or error processing.

IP.CPLM also specifies the number of additional CP seconds that an INTERCOM user can have after the session time limit (defined in the password file) is reached. IP.CPLM is adjusted if the user is executing a MUJ command.

IP.DBAL (3777B)

Installation-defined value for default batch access level. Set this value to 2 and IP.IACES to 1. The system assigns this value to any job that is not interactive, and the loader checks the value before loading system library resident programs. (The value for interactive jobs depends on the access level assigned in the password file.)

IP.CR (69D)

If set to 6, all BCD cards are read as if punched by a 026. If set to 9, all BCD cards are read as if punched by a 029. If set to 69, all BCD cards are read as if punched by a 026; however, if a job card or a 7/8/9 card has 29 punched in columns 79-80, all following BCD cards in that job are read as if punched by a 029, until a following 7/8/9 card changes the mode again. If set to 96, the inverse is true: 029 is default and job and 7/8/9 cards may switch to 026.

IP.CSET (IP.C64.1)

Using the IPARAMS symbols indicated, you can select one of two graphic character sets, CDC Scientific or ASCII. It can also independently select one of two character set sizes, 63 or 64 characters. The default character set is the CDC 64-character set. The IPARAMS modifications used to select each of the other three possible character sets are as follows:

ASCII 64-character set	IP.CSET	EQU	IP.C64.2
CDC 63-character set	IP.C63	EQU	IP.C64.1
	IP.CSET	EQU	IP.C63
ASCII 63-character set	IP.C63	EQU	IP.C64.2
	IP.CSET	EQU	IP.C63

The relationship chosen for IP.CSET, IP.C63, IP.C64.1 and IP.C64.2 must hold constant when all products referencing them are assembled for inclusion in a deadstart tape.

The character sets are described in detail in the NOS/BE 1 Reference Manual.

IP.C176 (0)

If nonzero, code to support model 176 systems is assembled. This option is automatically enabled if IP.819 is equated to one.

IP.ECSB (0)

If zero, the code to use extended memory is not assembled. If nonzero, the code to use extended memory is assembled and the extended memory installation parameters are activated.

IP.ENAP (0)

If zero (on 180-class mainframes), the central processor purges the instruction stack after executing an exchange jump (XJ), return jump (RJ), unconditional branch, UEM read instruction, or any branch outside of the instruction stack. Users can change the instruction stack purging status of the central processor on 180-class mainframes with the MACHINE control statement (refer to the NOS/BE Reference Manual).

If one, the processor also purges the instruction stack after central memory store and conditional branch instructions. Due to degradation caused by the additional stack purging, consider recompilation of existing binaries having code dependent on a particular stack length.

IP.IACES (11D)

Installation-defined size of the access level field. It should be identical to the INTERCOM definition of IP.IACES (refer to sections 12 and 28).

IP.IOLM (100B)

Installation-defined number of I/O seconds by which a job is incremented if it exceeds I/O time limit, as specified on the job statement or IP.SIOL, if needed for EXIT or error processing.

IP.IQD (6)

Input queue priority increment delay. The input queue priority is incremented by one every 2**IP.IQD seconds (0-11).

IP.IQPW (3)

Input queue priority weight (0 through 12). When a job is being considered for initiation, its effective input queue priority is P*2**(n-IP.IQPW)+A where P is the job statement priority, A is the age factor, and n satisfies the relation $4000_8 \leq$ IP.LVF*2**n $\leq 7777_8$.

The aging process does not allow the age factor to exceed a maximum value of 2^n-1. Thus, if IP.IQPW=0, a job with a higher job statement priority will always be initiated before a job with a lower job statement priority, regardless of the length of time the lower priority job has been waiting in the input queue. If IP.IPQW=12, job initiation is determined solely by age factor; job statement priority will not affect the choice. Selecting a value between these extremes allows both factors to be taken into consideration and provides a means for weighting one factor over the other.

IP.LINK (1)

Maximum number of links connected to this mainframe. If IP.LINK = 0, the linked 6000/7000 and the linked 6000/6000 command/display code will not be assembled. The following DSD overlays will not be assembled: 8YA, 8YB, 8YC, 8YD, 8YE, 8YF, 8YG, 8YH, 8YI, 8EB, 8EC, 8EE, 8EH, 8EJ, 8EP, 8YR, 8YU, 8YZ.

IP.LVF (70B)

Lowest fixed priority (2 through 7777_8). A fixed priority does not age and cannot be specified by a user (IP.LVF must be greater than IP.MPR). The value of IP.LVF also affects the processing of input queue priorities (refer to IP.IQPW), output queue priorities (refer to IP.OPRI), and job queue priorities. The calculation of a job queue priority includes a weighting factor as follows:

P*2**(n-6)*10_8

where P is the job statement priority and n satisfies the relation $4000_8 \leq$ IP.LVF*2**n 7777_8.

IP.MCPU (1)

Installation option to define maximum number of CPUs to be used by system. The value 1 produces the most efficient code for use on a single CPU. The system runs on a dual CPU machine, but uses only one CPU. The value 2 produces a variant of MTR which runs on a dual CPU machine using both CPUs or less efficiently on a single CPU machine.

IP.MECS (0)

Maximum number 0 to 7777_8 of 1000_8 word blocks of direct access extended memory that may be assigned in response to a job statement EC parameter, RFL statement, or MEMORY macro. This value determines whether sections of code are to be assembled within the system to handle extended memory allocation. This parameter must be nonzero if IP.ECSB is nonzero. IP.MECS should not be set equal to direct access total length as about 40K octal is used for storing system segments in extended memory.

In a multimainframe environment, the value of the EC parameter on the job statement is tested against IP.MECS only on the mainframe specified by the ST parameter on the job statement.

IP.MMS (100B)

Maximum mass storage limit, 1 through 377777_8, that may be specified by PRUs/100 (octal) on a LIMIT statement.

IP.MPPU (10D)

The maximum number (10 through 20) of peripheral processors in the configuration of any of the CMRs on the deadstart tape. A value of 20 allows execution on a 10-PP machine at the cost of reduced central memory availability. A value of 10 allows execution on a 20-PP machine at the cost of a reduced number of PPs available to the system.

IP.MPR (20B)

Maximum priority (1 through IP.LVF-1) a user can specify on his job statement. If a user specifies a higher priority, the default IP.SPR is used.

IP.MSCT (0)

Maximum decimal number of messages (1 to 4095) that may be entered into the dayfile by a single job. Only messages sent through MSG are counted. If zero, there is no maximum limit. Setting IP.MSCT $\neq$ 0 may cause some installation jobs to fail because of excess dayfile messages.

IP.MSLM (200B)

Installation-defined number of mass storage PRUs by which a job is incremented if it exceeds mass storage limit, as specified on the LIMIT statement, if needed for EXIT or error processing.

IP.MTL (77777B)

Maximum CP time limit in seconds, 0 to 77777_8, that may be assigned to a job. Both 0 and 77777 are considered infinite.

IP.NDEN (3)

Density for label and data on 1/2-inch 9-track tape, if not declared on REQUEST or LABEL statement (1=6250 cpi, 2=800 bpi, 3=1600 bpi).

IP.NDFS (1)

Number of dayfile copies on output. Up to 4095 may be specified.

IP.NJFL (20B)

$FL/100_8$ assigned to batch jobs when first assigned to a control point. The range is 1 to IP.MFL. The default value allows execution of job setup utilities.

IP.OPRI (0)

Specifies whether the size of a file affects its output queue priority. If IP.OPRI is zero, the priority is $P*2**(n-6)*100_8+1$ where P is the job statement priority and n satisifies the relation $4000_8 \leq IP.LVF*2**n \leq 7777_8$. If IP.OPRI is nonzero, the output queue priority is $P*2**(n-6)*100_8+2**n-1-S*2**(n-10)$ where P is the job statement priority, S is the file size in PRUs, and n is as previous described. If the file size exceeds 1777_8 PRUs, the IP.OPRI=0 is used.

IP.OQD (10B)

Determines period for incrementing priority of a job in the output queue. This period is 2**IP.OQD seconds. Legal values for IP.OQD are 0 through 13_8.

IP.PD (6)

System default print density in lines per inch. Legal values are 6 or 8.

IP.PFRP (5)

Default retention period in days for permanent files cataloged without explicitly defined retention periods. The range of values is 0 through 999.

IP.POSFL (5)

Field length/100_8 reserved for use by 1SO for requesting positive field length. Positive field length is not available to user jobs and can be considered part of CRM. Positive field length is allocated internal to the system for swapout use only. Range (4 to 10_8).

IP.PS (60D)

System default page size in lines per page. Legal values are in the range 16 through 255.

IP.PW (136)

System default page width in characters. Legal values are in the range 40 through 255. This parameter is currently used only by the common product set (compilers).

IP.RM (IP.HT)

Default recording mode of 844 and 885 disk packs.

IP.HT	Half-track recording mode
IP.FT	Full-track recording mode

NOTE

Full track recording mode may be used only on an 844 disk accessed by at least one 7154 or 7155 controller or an 885 disk accessed by at least one 7155 controller with a 170-class or 180-class mainframe.

IP.SCHDE (0)

Tape scheduling for 9-track units

0 Disables tape resource scheduling by density.

1 Enables tape resource scheduling by density. Job statement processing and all 9-track tape unit scheduling are based on user density requests. Request and label statements must match job statement density requests.

IP.SECS (0)

Default number of direct access extended memory blocks (1000 octal words) to be assigned to a job if not declared on job statement; range zero to IP.MECS.

IP.SEQ (0)

Job sequencer status after level 0 or level 1 deadstart

0 Do not start up job sequencer automatically during post-deadstart.

1 Start up job sequencer automatically during post-deadstart.

IP.SFL (50000B)

Default central memory field length (octal) to be assigned to a job if not declared; range 100 to IP.MFL.

IP.SIDLE (1)

If nonzero, code to support IDLE mode is enabled. Enable this code if the system checkpoint capability or full Status/Control register monitoring is desired.

IP.SIOL (0)

Default I/O time limit in octal seconds (0-77777B) to be assigned to a job if not declared on the job statement. A value of zero is considered infinite. If IP.SIOL is set to a value other than zero, assemble TDS with an I/O time limit of zero on the job statement for the EDITLIB (SYSTEM, RESTORE) job at EDITPRUF and the LDCMR job at LDCMPRUF. Set the I/O time limit on the archived retrieval job to zero (near 1PF.180).

IP.SMS (0)

If nonzero, the default mass storage PRU limit a job can use, divided by 100 (octal). All jobs, therefore, proceed as if a LIMIT statement with value IP.SMS were in the job deck. Refer to the LIMIT statement in the NOS/BE 1 Reference Manual. 1SI assembles with a type 7 error (refer to the COMPASS Version 3 Reference Manual, Error directory) when IP.SMS exceeds 7777B, but the resulting code is correct. IP.SMS must not exceed 377777B (17 bits).

IP.SPR (10B)

Default priority given to a job if no priority specified on job statement. Range 1 to IP.MPR.

IP.SPT (0)

If zero, no Scheduler performance execution statistics are returned. If nonzero, such statistics are returned.

IP.SRMS (0)

If nonzero, first mainframe to deadstart appears on the operator option matrix during deadstart. Also, certain key code for RMS sharing is controlled with this parameter; thus, if shared RMS (PL1H) is not installed, IP.SRMS should be zero.

IP.STL (100B)

Default central processor time limit in octal seconds (00 to IP.MTL) to be assigned to a job if not declared on the job statement. Values of 0 and 77777B are considered infinite.

IP.TCPUB (4)

CPU time for CPU A is accumulated at a rate that is IP.TCPUB/4 times greater than actual time used. The intention of this parameter is to equalize the time that will be accrued on either CPU of a 6700 or model 74-2x. It could also be used on a single CPU. An installation that has both a model 73 and a model 74 could use IP.TCPUB 8 on the model 74 to equalize the effect of the time limit on either machine.

IP.TDEN (2)

Density for both label and data on 1/2-inch 7-track magnetic tape if not declared on LABEL or REQUEST statement (0=556 bpi, 1=200 bpi, 2=800 bpi).

IP.TYPE (6600)

Determines the type of central processor to be used by the operating system (6600 -- model 175, 176, 740, 750, 760, 865, 875, 990, or model 74; 6400 -- model 172, 173, 174, 720, 730, 740, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, or model 71, 72, or 73; or 6500) for generation of optimal code. Acceptable values are 6400, 6500, and 6600.

IP.UP (10B)

Determines the permissions granted to a user who has specified the installation-defined universal password. This is a 4-bit field in which each nonzero bit signifies the type of permission granted.

Bit Position	Permission Granted
3	Control
2	Modify
1	Extend
0	Read

IP.US (0)

When you reserve space (user slot) in the Permanent File Catalog for information to be saved with each permanent file, this parameter is the space length in central memory words. 20_{10} is the maximum value supported.

IP.819 (0)

If nonzero, code to support the 819 RMS subsystem is activated. LDCMR errors occur if this code is activated without installing code from PL1G.

IP.YMD (MDY)

Micro which shows format of date to be typed in at deadstart. The six possible permutations of the letters MDY constitute the range of this parameter.

MODEL (74)

Micro, used by the product set members for optimal code generation, whose value is the CDC CYBER 70 or CDC CYBER 170 model number corresponding to the type of central processor for which code is to be generated and optimized. Acceptable values are 71, 72, 73, 74, 171, 172, 173, 174, 175, 176, 720, 730, 750, 760, 810, 815, 825, 830, 835, 840, 845, 850, 855, 860, 865, 875, or 990. The recommended value for a 6400 or 6500 is 73, and the recommended value for a 6600 or 6700 is 74.

OS.ID (NOS/BE 1.5)

System identification micro used by the product set members for displaying the operating system name and version number in generated program binaries.

TAPE PROCESSING INSTALLATION PARAMETERS (CIOCOM)

The default values of the CIOCOM configuration parameters are defined with the CEQU or CMICRO macros, so that you can insert all modifications at one given place. The CEQU and CMICRO macros are used to define variables conditionally. As they are effective only if the variables have not been previously defined, any modifications should precede them.

Installation parameters specifically oriented to tape processing are defined within the COMDECK CIOCOM. CIOCOM is listed in the routine CMR. Assigned (default) values, other tested values, and descriptions are as follows. Make changes to default values at CIOCOM.6 in an update of PL1A.

IP.CBKSP (1)

If one, controlled backspace is available in all controllers for 65x drives; if zero, it is not installed.

IP.NBCD (0)

9-track default conversion mode (0=ANSI, 1=EDCDIC)

IP.NBRK (0)

System noise record usage on 65x tape drives only. If zero, system noise records are used in write recovery at densities other than 1600 bpi. If one, they are not used. It is recommended that you run with noise bracketing enabled to take advantage of increased reliability on tapes which are not destined for interchange, and that users who are creating tapes for interchange purposes include an IP parameter with their tape requests.

Study has shown that the use of noise brackets on phase encoded tapes has not increased their reliability. For this reason, noise brackets are never written on phase encoded tapes.

IP.NTCN (2)

Number of tape channels.

IP.NOISE (3)

Maximum decimal number of 12-bit bytes in a noise record on 7-track S and L tapes or 9-track conversion mode (S-format) magnetic tape. A record less than or equal to IP.NOISE is discarded.

IP.NOIS9 (5)

Maximum decimal number of 8-bit bytes in a noise record for packed mode on 9-track tapes. A record less than or equal to IP.NOIS9 is discarded.

IP.PTCN (13B)

Primary tape channel number. Used for internal purposes.

IP.RCYC (3R000)

Retention cycle (0 through 999) for calculating tape label expiration date when no retention cycle is given; 999 indicates permanent retention. The address field of the symbol definition should contain 3Rxxx where xxx defines the retention cycle; leading zeros need not be written.

IP.RPE1 (12D)

Total decimal number of read parity retries on a single record (must be less than 60).

IP.RPE2 (8)

Decimal number of read parity retries accomplished by backspacing over the previous three records, then reading forward in an attempt to recover (IP.RPE2 must be less than IP.RPE1).

IP.TSG (2617B)

Tape scheduling options are as follows:

Name	Bit	Value	Definition
S.AUTO†	0	1	Enable automatic tape assignment according to LABEL or VSN specification.

†These bits are enabled in the default value of 2617B.

Name	Bit	Value	Definition
S.URES†	1	1	Enable job scheduling based on job statement reservation.
S.PRES†	2	1	Enable prestaging features (the VSN preview of the P display).
S.2LBP†	3	0	Only ANSI labels are accepted and written.
		1	Two label formats (ANSI and 3000) are defined.
	4	1	Unused.
	5	1	Unused.
S.SCUL	6	1	Write-enabled, unlabeled tapes will be considered as usable for automatic assignment as scratch tapes.
S.SCEL†	7	1	Write enabled expired labeled tapes will automatically be considered for assignment as scratch tapes.
S.SCBL†	8	1	Write-enabled blank labeled tapes will automatically be considered as scratch tapes.
S.PREA	9	0	Give warning if tape job has no VSN information.
		1	Preabort such jobs.
S.OCJI†	10	0	Job initiation is based on tape drive availability; total demand cannot exceed number of drives logically available.
		1	Job initiation allows tape drive overcommitment.
S.UEOJ	11	1	Unless specified otherwise on REQUEST or LABEL statements, all tapes are unloaded at end of job.
S.PSON	12	1	Prestaging feature set on at deadstart time. This is equivalent to the STAGE ON typein.
S.TSEC	13	0	Tape security off. Specification of RING or NORING causes requested action.
		1	Tape security on, installation default active. Specification of RING or NORING causes requested action.
S.TRDO	14	0	Establishes installation default of NORING.
		1	Establishes installation default of RING.
	15-16		Unused.
S.NOOR	17		Operator cannot override VSN card.
S.DBUG	18		Enable label debug code (4LB,4LC).
	19-20		Unused.

† These bits are enabled in the default value of 2617B.

Tape scheduling options that you can select are implemented by the use of conditionally assembled code. The bits in IP.TSG are tested at assembly time to determine the exact nature of the programs that compose tape scheduling. For example, bit S.SCBL in IP.TSG governs the automatic scratch status of blank labeled tapes. If the bit is on, blank labeled tapes are considered scratch without operator intervention; if the bit is off, scratch status is not granted automatically.

The bits in IP.TSG can be divided into the three general categories of automatic assignment bits: prestaging bits, overcommitment bits, and miscellaneous bits.

AUTOMATIC ASSIGNMENT BIT

You can select automatic assignment by setting bit S.AUTO. With S.AUTO set on, a specific tape will be assigned automatically when the specific tape is mounted.

AUTOMATIC SCRATCH STATUS

Three other bits are related to automatic assignment. They are bits S.SCUL, S.SCEL, and S.SCBL. When set, each bit determines a specific type of tape to be considered automatically as a scratch tape. If all three bits are off, the only tapes treated as scratch are those specifically designated by the operator with the command SCRuu (where uu is the EST ordinal).

A job specifies *MT or VSN = SCRATCH in the request for a scratch tape. If any automatic assignment is turned on (bit S.AUTO is set), the system will try to assign a scratch tape automatically to the job. The tape must be mounted on a ready unit with a write ring in place, it must also be designated as scratch as described above, and it must meet the following qualifications.

Tapes designated as scratch by the operator

Unlabeled tapes if bit S.SCUL is on

Tapes with expired labelsd if bit S.SCEL is on

Tapes with blank labels if bit S.SCBL is on

PRESTAGING BITS

Unit Reservation

Bit S.URES controls the necessity of job statement tape parameters, without which overcommitment and deadlock prevention are meaningless and prestaging will not function.

Prestaging

The prestaging option is assembled if bit S.PRES is set. If this option is on, a prestaging buffer is assembled in CMR; its length is N.VRNBUF*6 (release value gives a 171B word buffer). You can change symbol N.VRNBUF in CMR to change the size of the buffer.

Complete VSN information cannot be obtained for jobs making internal tape request or using tape file names repeatedly.

If bit S.PSON is on, it sets up CMR as if STAGEON had been typed after a normal deadstart. Deadstart recovery preserves the current setting of the STAGEON/STAGEOFF switch.

When bit S.PREA is set and prestaging is on (operator entered STAGE,ON or bit S.PSON set from deadstart tape), all jobs that specify tapes on the job statement but do not supply VSN information for all tape files requested are aborted.

OVERCOMMITMENT BIT

Bit S.OCJI determines whether or not tape drives will be overcommitted. If the bit is off, the total number of tape drives required by all jobs executing at a given time (as determined by job statement tape parameters) cannot exceed the total number of tape drives at the installation. If bit S.OCJI is on, tape drives are overcommitted; the total tape requirements of executing jobs can exceed the total number of tape drives at the installation. Deadlock is prevented by an algorithm calculated each time a tape is assigned.

MISCELLANEOUS BITS

Two-Label Processors

If, in addition to the ANSI label processor 4LB, 3000 Series (Y) labels are to be processed, set the bit S.2LBP on to allow use of the alternate label processor 4LC.

EOJ Tape Unload

Bit S.UEOJ causes 1EJ to unload nonscratch tapes at end of job. If you encounter any problems when trying to unload the tape, such as tape not ready, the unload attempt is ignored. This differs from the SAVE (SV on REQUEST statement or X=SV on LABEL statement) unload processed by 1EJ; 1EJ issues a message that problems exist and continues trying until the operator types in GOuu.

Operator Cannot Override VSN

With bit S.NOOR off, the operator can assign a tape with a VSN different from the VSN specified by the job; with S.NOOR on, a different VSN is not allowed.

Label Debug

Bit S.DBUG controls debug code in 4LB and 4LC; use of this bit is not the normal mode of operation. This debug code produces many messages which show the calls to and returns from the label processors. Such messages may cause other more informative messages to be overwritten.

OPTION DEPENDENCIES (IP.TSG)

The figures below show dependent bits. Each bit name shown cannot be turned on (turning it on will have no effect) unless all bit names below it are on. The three groups of bits are independent of each other. Miscellaneous bits (S.2LBP, S.UEOJ, S.NOOR, and S.DBUG) are independent of each other and of the bits shown below.

Auto Assign Dependencies

<table>
<tr><td>S.SCUL</td><td>S.SCEL</td><td>S.SCBL</td></tr>
<tr><td colspan="3">SAUTO</td></tr>
</table>

Prestage and Overcommitment Dependencies

<table>
<tr><td>S.PSON S.PREA</td><td></td></tr>
<tr><td>S.PRES</td><td>S.OCJI</td></tr>
<tr><td colspan="2">S.URES</td></tr>
</table>

For example, S.PREA is dependent on S.PRES and S.URES but not on S.OCJI.

Tape Security Dependencies

<table>
<tr><td>S.TRDO</td></tr>
<tr><td>S.TSEC</td></tr>
</table>

IP.WEC

Hardware write error correction (applies only to 6250-cpi density). Default value is 0. If IP.WEC is zero, the system allows certain types of single-track errors to be written that can be corrected when the tape is read (on-the-fly correction). This is the recommended setting because it provides efficient throughput, error recovery, and tape usage when writing GE data on media suitable for use at 3200 fci or 6250 cpi.

When IP.WEC is one, the system invokes standard error recovery processing when an on-the-fly error occurs when writing a GE tape. The system erases the defective portion of tape, thereby reducing the amount of data that can be stored on the tape. Only tape which is suitable for recording at 6250 cpi should be used when this mode of operation is in effect.

NOTE

Users can override the installation parameter through the REQUEST and LABEL statement parameters EEC (enable error correction) and IEC (inhibit error correction). Refer to the NOS/BE 1 Reference Manual.

Refer to part III for a cross-reference listing showing the routines that reference each IPARAMS and CIOCOM symbol.

MISCELLANEOUS INSTALLATION VARIABLES

PASSWORD DEFINITIONS FOR THE SYSTEM DAYFILE

You may protect the system dayfile from unauthorized attaches by defining two passwords. The two passwords provide RD and XR permissions for the file. These passwords are defined as micros locally in both TDS (PL1A) and 1DF (PL1B).

The two micros are the following.

(XR permission) SDFXR

(RD permission) SDFRD

The default permissions defined in TDS and 1DF afford no protection for unauthorized attaches. Redefine micros SDFXR and SDFRD to protect the dayfile. Consult a listing of TDS or 1DF to review the default definitions and to determine where new definitions must be inserted.

To change the SYSTEM dayfile passwords, make the same changes in both TDS and 1DF. When these changes are made they must be coordinated with an initial deadstart. When TDS catalogs the dayfile after an initial deadstart the passwords thus defined remain in effect.

MACROS TO DEFINE SPACING CODE ARRAYS FOR JANUS

580 printers equipped with programmable format control (PFC) make use of software-defined arrays instead of format tapes to specify spacing codes. Two pairs of arrays (each containing a 6-line-per-inch array and an 8-line-per-inch array) are specified in PP routine 1IU. The first pair defined is the default pair, and is used for SC=0 and SC=1 on the ROUTE control statement. The second pair is the alternative pair, and is used for SC=2. Space exists for 61 installation-defined array pairs, corresponding to SC values 3 through 77_8. New arrays may be added by the use of the DPFC macro.

```
DPFC      *vcode*,*param*
```

Parameter	Description
vcode	Specifies the number of line per inch for the array being specified

Value	Description
V6	6-line-per-inch array.
V8	8-line-per-inch array.

Parameter	Description
param	Defines the actual array. This can be specified using letters A through L, O, and X. Letters A through L define channels 1 through 12, respectively. A indicates the beginning of the array, O indicates the end of the array, and X indicates no channel in that position. You must specify a letter for each line in the form.

Each array should also conform to the following criteria.

- It must begin with an A.
- It must end with an O. This is an end-of-array terminator that is not counted as a line.
- It must not be longer than 132 characters plus end-of-array terminator (6-line-per-inch array) or 176 characters plus end-of-array terminator (8-line-per-inch array).
- It must contain each channel specified at least once in the array (L specifies the bottom of form).
- Arrays must be specified in pairs; one 6-line-per-inch array and one 8-line-per-inch array. Either array may be specified first.

Example:

```
*B DPFC.1
       DPFC          *V6*,*AXBXCXDXEXFXGXHXIXJXKXBXCXLO*
       DPFC          *V8*,*ABCDEFGHIJKLO*
```

MACRO TO DEFINE THE EC PARAMETER ON THE JOB STATEMENT

You can change the syntax of the EC parameter field on the job statement through the OPTION macro in common deck 2VJCOM.

```
OPTION     type,spec,mode,defl,base
```

Parameter	Description
type	Specifies the parameter field name; EC is required.
spec	Specifies the format of the EC field.

spec	Description
0	EC can be specified without a value. A default value, defl, is supplied.
other	A value is required with EC; EC by itself is ignored.

Parameter	Description
mode	Specifies initial assignment of extended memory field length at beginning of the job.

mode	Description
REDUCE	Extended memory field length is not assigned.
Other or omitted	Extended memory field length is assigned.

defl	Specifies the default value (in multiples of 1000 octal) that is supplied when EC appears without a value. It should not exceed IP.MECS. It has no effect when spec is nonzero.
base	Specifies the base of the value following EC.

base	Description
DECIMAL	Value is decimal.
Other or omitted	Value is octal.

On the released system, the OPTION call is as follows:

```
OPTION                  EC,1,RFL
```

This indicates that an octal value must follow the EC parameter on the job statement, and that the extended memory field length will be assigned at the start of the job.

CMR CONFIGURATION PARAMETERS (CMRIP)

The default values of the CMR configuration parameters are defined with the CEQU or CMICRO macros, so that you can insert all modifications at one given place. The CEQU and CMICRO macros are used to define variables conditionally. Since they are effective only if the variables have not been previously defined, any modifications should precede the release definitions.

Symbols can be defined by EQU or CEQU except for IP.SYSL1, IP.SLIB, IP.VER and IP.SYSE, which are micros and must be defined by CMICRO.

All the CMR configuration parameters are grouped together near the beginning of CMR.

Tailor general parameters to suit the needs of your installation; default values are shown in parentheses.

Make changes as insertions after CMRIP.1 in an update of PL1A.

CMR.PD	(IP.PD)	System default print density in lines per inch. The job print density is set to this value when initiated.
CMR.PS	(IP.PS)	System default page size in lines per page. The job size is set to this value when initiated.
CMR.PW	(IP.PW)	System default page width in characters. The job page width is set to this value when initiated.

L.ELST	(24)	Length of error logging status table. When executing on a 6000 series machine, it is recommended that this parameter be set to zero. Error logging cannot be performed on 6000 series machines.
IP.ELST	(0)	Error logging mode. Value is displayed on the main operator option matrix during deadstart. On a model 990 mainframe, the default is dedicated. Value — Description 0 — Nondedicated 2 — Dedicated
L.EST	(40B)	Length of equipment status table ($\leq 1000_8$). Only RMS devices may be placed in the EST above 77_8.
L.INS	(0)	Length of installation table. Size, definition, and usage of an installation table is completely controlled by the individual site. No NOS/BE product set program makes reference to the installation table.
L.FNT	(2200B)	Length of file name table.
L.LDIS	(0)	Length of linked station displays, in lines. Range of values (0 to 420). Standard NOS/BE screen size is 42 lines. A value of zero means the length of the standard screen. The value of L.LDIS is used by DSD only in message code MC.LDIS when IP.STEX is set to nonzero.
L.SEQ	(10B)	Length of the sequencer table. To use the sequencer, the value must be (2* number of jobs to be run)+2. If defined as zero, the sequencer cannot be used.
L.IDT	(40B)	Length of ID table. Must be nonzero multiple of 8.
LE.DFB00	(400B)	Size of system dayfile buffer may be less than 100_8; if not, then it must be an even multiple of 100_8.
LE.DFBXX	(77B)	Size of control point dayfile buffers.
LE.CERFB	(46B)	Size of hardware error file buffer. Subject to same limitation as LE.DFB00.
IP.ECSTP	(0)	Type of extended memory. If zero, the flaw area that exists for ECS I is not considered for allocation; if nonzero, the entire FL of extended memory is considered for allocation. On an 865 or 875 mainframe, a value of one indicates that ESM is used for extended memory and a value of two indicates that UEM is used. Note that on a 180-class mainframe, a value of one still results in the use of UEM for extended memory.

IP.MFL	(140000B)	Maximum amount of central memory field length that may be assigned to a user job. A user cannot request more than IP.MFL field length on a job statement or with MEM or RFL. For additional information, refer to Scheduling Parameters in this section. IP.MFL must not exceed 377700_8.
N.BRKPT	(10B)	Maximum number of CPMTR breakpoints (for DEBUG only).
N.CP	(15)	Number of control points (1 to 15 decimal).
N.DEVICE	(3)	Number of controllers for allocatable devices; one for each 841, 844, or 885 disk pack controller (which may drive more than one disk pack unit). This parameter only has an effect if no RMS devices are assembled in the EST.
N.RBR	(3)	Number of record block reservation tables; normally one for each 841 unit, one for each 844-21 disk pack unit, two for each 844-41 double-density disk pack unit, and two for each 885 disk unit. This parameter only has an effect if no RBR cards are assembled.
N.RQS	(40)	Number of request stack entries.
N.VRNBUF	(20)	Number of entries in tape VSN buffer. Each entry is six words long and represents one line of job tape VSN information in the P display.
N.SPRPP	(1)	Number of PPs that are to be reserved for stack processor. In all cases, N.SPRPP must be at least one. If any dual access devices are defined, N.SPRPP must be at least two. Always use the minimum value (1 or 2) if there are only seven or ten PPUs.
IP.FTHRL	(40B)	Default lower limit of free FNT entries. When the number of free FNT entries falls below this value, the system enters an FNT space critical condition. Specify a multiple of 10_8; the units position is truncated and assumed to be 0.
IP.FTHRU	(140B)	Default upper limit of free FNT entries. When the number of free FNT entries rises above this value, the system clears the FNT space critical condition. Specify a multiple of 10_8; the units position is truncated and assumed to be 0.
IP.DCT	(0)	Default controller type for 844 controllers.

Value	Description
0	Half-track (7054) controller.
1	Full-track (7154) controller.
2	Full-track (7155) controller.

IP.SYSL1	(NOS/BE 1.5)	System label (up to 20 characters); the first character must be blank.

IP.VER	(∧RELEASE∧∧)	System version identifier (up to 10 characters).
IP.SYSE	("DATE")	System generation date (up to 10 characters). These 10 characters are changed to type of deadstart and time when a deadstart recovery is done.
IP.SLIB	(CMRLIB)	Name of the library containing the segments and CMRTEXT corresponding to this CMR.
L.APF	(64)	Length of APF table (2-word entries) $12 \leq$ L.APF ≤ 8190.
N.SETS	(2)	Maximum number of device sets which may be mounted at any one time. The mounted set table (MST) will have a length of N.SETS*LE.MST. The range for N.SETS is 1 through 63.
N.VDDT	(3)	Maximum number of permanent packs for which jobs can be swapped out. The dismountable device table (DDT) will have a length of N.VDDT + the number of RMS EST entries.

System control point parameter:

N.SBSYS	(0)	Maximum number of subsystems that may be defined in the subsystem control table (minimum of 4). If zero, SCP code will not be assembled in CMR.

Scheduler parameters (refer to Scheduling Parameters in this section):

L.ECSSWP	(3)	Extended memory swap mask. The value of this symbol forms a mask of bits that control job swapping when extended memory is available.

Bit	Meaning When Set
0	Swap any INTERCOM or graphics job to extended memory at end of command.
1	Swap any batch job to extended memory at end of time quantum or if waiting for memory.
2	Swap any INTERCOM or graphics job to extended memory (overrides bit 0).
3	Swap any batch job to extended memory (overrides bit 1).
4	Swap any job except MUJ to 819 (overrides bits 0-3).
5	Reserved.
6	Swap a job's direct access extended memory and CM field lengths simultaneously. (This is effective only if IP.ECSW is nonzero; refer to Extended Memory Installation Parameters.) A job's extended memory is always swapped to mass storage.

		If bits 0 through 4 are zero, only MUJ jobs and INTERCOM or graphics jobs at end of time quantum or waiting for memory will go to extended memory. Set bit 4 only if an 819 is available. Set bit 6 only if the system has two separate channels to the mass storage devices that can hold swap files.
L.SCHJCA	(20B)	Length of job control area. Must be a multiple of 8. (Needs to be changed only if new classes are added.) L.SCHJCA can be redefined by means of the COMPASS pseudo-op SET.
L.SCHJDT	(400B)	Length of job descriptor table. Must be a multiple of 8.
AFL.BAS	(0)	Anticipated field length/1000B used when INTERCOM is not up.
AFL.INT	(30B)	Anticipated field length/1000B used when INTERCOM is up (regardless of whether or not RDF is up).
AFL.RMF	(22B)	Anticipated field length/1000B used only when RDF is up.

LOGICAL ID TABLE (LID)

The Logical ID Table (IDT) in CMR contains the mainframe Host ID (HID), the associated Logical IDs (LIDs) and the Physical Link ID (PID) of each currently linked mainframe of a multimainframe network. The last character of the HID in any multimainframe network must be a unique letter. One and only one HID can exist for a given mainframe IDT. Up to 58 logical IDs can exist. L.IDT must be nonzero and a multiple of 8. The default HID is MFA. Change the HID by the following.

```
    *INSERT,CMRIP.1
    HOSTID CMICRO, (xxx)          where xxx is the desired HID.
```

Add logical IDs by the following.

```
    *INSERT,LID.1
             LID xyz              where xyz is the desired logical ID.
```

SYSTEM SECONDS

At end of job or when SUMMARY is executed, total system seconds is calculated and reported along with other job accounting in dayfile messages.

System seconds is expressed mathematically.

```
SS = CP*AW + IO*BW + CM*CW + EC*DW

    CP is CPU A plus CPU B time in seconds.
    IO is I/O time in seconds.
    CM is central memory seconds.
    EC is extended memory seconds.
    AW, BW, CW, and DW are installation selected weighting constants.
```

Central memory and extended memory seconds are similar and can be expressed mathematically.

```
(CP*EW + IO*FW) * FL

    CP is CPU A plus CPU B time in seconds.
    IO is I/O time in seconds.
    EW and FW are installation selected weighting constants.
```

Terms and their sum (SS) are calculated by CP monitor when a PPU requests the M.ICE function, EX.SS subfunction. The elements CM and EC are calculated by the CP monitor routine PACKAGE each time central memory or extended memory field length changes, or when system seconds is requested by a PPU.

The general format of the dayfile messages which show job accounting is the following:

$CPA	Raw CPU A time in seconds	Weighted CPU A time
$CPB	Raw CPU B time in seconds	Weighted CPU B time
$IO	Raw I/O time in seconds	Weighted I/O time
$CM	CM seconds in kiloword seconds	Weighted CM seconds
$EC	Extended memory seconds in kiloword seconds	Weighted extended memory seconds
$SS		Sum of weighted terms
$PP	Raw PP time in seconds	Today's date

PP time is reported although it is not added into the system seconds total.

If IP.TCPUB is not equal to 4, then the figure referred to as raw CPU A time in seconds is itself an adjusted time. It is (seconds *IP.TCPUB/4).

WEIGHTING CONSTANTS

Weighting contants, mentioned under the System Seconds, have the following values in the system release version.

```
AW = 1
BW = 1
CW = .061035156250 (1000./40000B)
DW = .030517578125 (1000./100000B)
EW = .001
FW = .001
```

These values are not intended to be fixed or necessarily optimal for any individual installation. The weighting constants are released with these values so that installations can gather meaningful statistics in order to adjust the values at a later time. Note that CP and IO weighting constants AW and BW have the value of 1 in order to report actual time. CM and extended memory seconds weighting constants CW and DW cause the weighted values to be in 40K and 100K octal units, respectively.

Core seconds weighting constants EW and FW have the value of 0.001 and cause CM and extended memory second values to be in 1000 decimal or kilowords seconds. The ratio of 1 to 1 is used even though you must determine the best ratio for your job mix.

AW, BW, CW, and DW are defined at the end of CP.SS in CP MTR (near CRESCH.213), EW and FW are defined at the end of PACKAGE in CP MTR (near CRESCH.373).

You should not charge the user for PP time. A significant portion of PP time is system overhead not specifically requested or desired by the user. PP time will not necessarily be constant for the same job across several runs because PP time used by a job is dependent upon system activity and configuration. For example, PP time accrued by a user job will vary depending on the residency of the PP routines. Additionally, PP time charged will vary depending on the system activity when the job is run.

TABLE STRUCTURES

Establishing a CMR for an installation requires inserting information about the CMR configuration parameters and tailoring the EST, RBR, and FLAW tables. Up to 64 different CMR configurations, each with unique EST, RBR, and FLAW tables, may be placed on the deadstart tape.

EQUIPMENT STATUS TABLE

The EST may be tailored to any configuration by using the macros described in this section. Its size may be greater than or equal to the number of hardware units present in the configuration. However, it may not exceed 777 (octal) since an EST ordinal must be no more than 9 bits. Since the first word of the EST cannot be used, the first equipment ordinal is 001. Only RMS devices may have an EST ordinal greater than 77_8.

The CMR tables are defined by the TABLE macro. The sequence of the macro calls defines the sequence of the tables generated in CMR. You can alter this sequence, but you must observe the following constraints.

- Origins of CST, EST, FNT, ITABL, DAT, RMSBUF, and STG tables must be located under 10000 (octal).

- The RQS table must be located under 20000 (octal).

EQUIPMENT CONFIGURATION

The EST macro defines the equipment in the configuration and the attributes associated with them. The actual parameters for the EST macro call are position independent and are given as a list of keyword=parameter fields separated by commas.

The EST macro causes a one-word EST entry to be constructed each time the macro statement occurs. For RMS devices, if no EST macro with the same type, channel 1, channel 2, and controller has been assembled, a DST and corresponding DAT entry will be constructed. If channel 2 is not blank, a second set of DST and DAT entries will be constructed for a dual access configuration that allowed for 841, 844, or 885. A DDT entry is also constructed for RMS devices, and an MST entry as well if MASTER is specified.

Make EST macro entries at EST.1 in a PL1A update. The macro format is the following.

```
dt  EST         keyword=xxx,keyword=yyy...
```

Parameter	Description
dt	Device type mnemonic; if dt is omitted, all parameters but ESTO are ignored.

Keywords include the following.

Keyword	Description
NAME	Device identifier; any combination of up to 6 letters or digits (must be unique for each RMS unit; used to map RBRs).
ESTO	EST ordinal (default is previous device count plus 1). The value must be larger than the previous device count.
CH	Channel numbers; multiple channels are given CH=(C1,C2,...).
EQP	Equipment number or display synchronizer number; multiple equipment numbers may be specified, such as EQP=(EQ1,EQ2,...).
UNIT	Unit number; if an expander is being used with 844-21 disk packs, the unit number can be two digits. If 885 disks are used, unit numbers are two digits between 40 and 57 (octal).
UNITS	Number of units defined; default=1 (automatically incremented as more are defined) not allowed for disk.
MUX	Multiplexer subtable index (for INTERCOM 4 only) or extended memory buffer number (if ECS link is used for multimainframe communication).
SN	Setname; defines set membership of RMS device.
VSN	Volume serial number of particular RMS device.
NF	Used on master device only to specify the maximum number of permanent files that the PFD for the set can hold (PFD will be slightly larger). NF is 1 through 16000. Default is 320 times the number of devices in the device set. NF=n determines the number of subdirectories (NSD); that is, the number of hash points and the number of pages per subdirectory

Keyword	Description
	(PPSD) when ADDSET calculates the size of the permanent file directory. ADDSET selects a value for NSD from the range of prime numbers 1 through 61. PPSD must be integral power of 2. To obtain the maximum number of hash points (NSD=61) in the PFD, select values for n approaching one of the following thresholds. 1952 3904 7808 15616 When the value for n surpasses one of these thresholds, the value for PPSD doubles, and the value for NSD reverts to 31.
NM	Used on master device only to specify the maximum number of members permitted in this set; default is equal to number of members in EST plus 5.
FC	Applies to printers, specifies an optional forms code (equivalent to operator's DSP type-in); value equals any two alphanumeric characters.
EC	Applies to printers, specifies external character set (type of print train on the printer); values include the following. A4 ASCII 48-character set (596-2) B4 BCD 48-character set (596-3) A6 ASCII 64-character set (596-4 or 596-5) B6 BCD 64-character set (596-1) A9 ASCII 96-character set (596-6)
MOD	Specifies device attributes; multiple attributes are given as MOD=(value1,value2,...) where value is one of the following keywords. For all devices: OFF For RMS devices, values are: FREE SYS System files may reside on device. PF Permanent files may reside on device. QUE Queue files may reside on device. SHAR Drive is shared between mainframes (applies only to 844s and 885s). IDLE Drive is initially idled.

Keyword	Description
	For tapes, values are:
	ATS — Specifies 67x tape subsystem.
	GCR — Used in conjunction with ATS parameter to indicate an ATS unit with 6250-cpi density capability.
	BID — Used in conjunction with the MTS parameter to indicate Block ID capability for 66x units (for example, MOD=(MTS,BID).
	MTS — Specifies 66x tape subsystem.
	MMTC — Specifies 65x tape controller.
	6684 — Specifies channel converter.
	For printers:
	PFC — Printer is equipped with programmable format control (applies only to 580 printers).
	For distributive data paths (DDPs), specify one of the following values:
	D135 — Specifies DDP type DC135.
	D145 — Specifies DDP type DC145.
	An extended semiconductor memory low speed port which is connected to a CYBER channel is functionally equivalent to a DDP. These low speed ports should be defined in the EST as a DDP type DC145.
	ED EST MOD=D145,...
MASTER	Specifies attributes of set for which this is the master device; multiple attributes are given as MASTER=(M1,M2,...)
	SYS — System files may reside on set.
	PF — Permanent files may reside on set.
	QUE — Queue files may reside on set.
	SCR — Scratch files may reside on set (system default set).
TYPE	Specifies the controller type for the previously specified channel and equipment numbers (in one-to-one correspondence if multiple parameters are included); specify multiple types as TYPE=(TY1,TY2,...).
	7054 — Specifies 7054 controller.
	7154 — Specifies 7154 controller.
	7155 — Specifies 7155 controller.

Use this entry only for 844 and 885 devices; it is ignored for other devices. If TYPE is not defined, 7155 is the default value for 885 disks, and the value of IP.DCT determines the default for 844 disks. If TYPE is not defined for the second, third, or fourth access of a multiple access disk, the last defined type is used.

NOTE

An attempt to redefine TYPE in a subsequent EST with identical CH and EQP attributes is ignored and the original definition is used.

CO

Specifies the RMS controller status for the previously specified channel and equipment numbers (in one-to-one correspondence if multiple parameters are included); multiple controller status options are specified as CO=(option$_1$, option$_2$,...option$_n$)

Options may be one of the following.

ON	Controller autoloaded and used.
OFF	Controller autoloaded but not used.
LOCK	Controller not autoloaded and cannot be used until controlware has been reloaded.

If CO is not defined, the default is ON.

NOTE

Any attempt to redefine CO in a subsequent EST with identical CH and EQP attributes is ignored and the original definition is used.

If all of the controllers through which a device is accessed are OFF or LOCKED, the EST entry for all devices whose status is not already IDLE will have their status set to IDLE and a warning diagnostic is issued.

Keyword	Description
SO	Specifies the stack request scheduling option for RMS devices. SEL — Specifies optimization of requests within each unit and unit selection scheduling. SEEK — Specifies overlap seek optimization in addition to SEL optimization. FIFO — Sends stack requests to the device driver in the order received. The default value is SEEK. This parameter is not allowed for 819 devices. Specify this parameter for the first device on the channel only. If specified for more than one device on the channel, the first specified SO option is the default for all devices on that channel. Other specified options for subsequent devices are ignored.

The device type may be any of the following.

AH	819 disk drive	LR	Line printer (580-12)
AJ	885 disk drive	LS	Line printer (580-16)
AM	841 disk drive	LT	Line printer (580-20)
AY	844-21 disk drive	MT	Magnetic tape (657, 667, 677)
AZ	844-41 disk drive	NC	380-170 Network Access Device (NAD)
CC	6683 satellite coupler	NT	Magnetic tape (659, 669, 679)
CR	Card reader (405)	CP	Card punch (415)
CS	791 LCC mux	DS	Console display
CX	ECS link	SC	6673/6674 wide-band mux
ED	Distributive data path (DDP) and ESM low speed port	DC	6671 low-speed mux
FE	255x Front End	YC	6676 low-speed mux
LQ	Line printer (512)	RM	Two-port mux

NOTE

The shared attribute is not specified via the EST macro for RMS controllers. This is not necessary, because the software assumes that any controller may be shared.

The AH device type can have only the attributes ON, FREE, PF, IDLE, and OFF. It cannot have the system device attribute (SYS) nor can it be a MASTER device. Also, it must be a member of a public set.

Parameters defining hardware configuration are required (CH, EQP, UNIT). RMS devices also require NAME, VSN, and SN, or only NAME if MOD=FREE. Do not use BSSZ to create spaces in the EST; use the ESTO parameter.

NOTE

A mixed 844 RMS controller configuration (7154 or 7155 full-track controllers and 7054 half-track controllers) sharing 844 drives is allowed but must be used with caution. This configuration must be operationally limited to data recorded in half-track mode to prevent serious performance degradation. This degradation occurs if full-track recording operations are performed through the 7054 half-track access.

All numeric parameters are assumed octal unless otherwise specified. Multiple parameters must be enclosed in parentheses. Channel numbers must be set up in order of precedence, except for RMS devices where channels should be specified in ascending numerical order.

NOTE

Before using a continuation card, information must be punched through column 72 of the previous card and the continuation card must begin with a comma in column 1.

In the following examples, the device type begins in column 1, the macro name in column 11, and the parameter string in column 18.

```
AY        EST    NAME=844D,CH=2,EQP=0,UNIT=3,ESTO=10,MOD=(OFF,QUE,SHAR),
,VSN=SHARIO,SN=IOQUES,MASTER=(QUE,SCR),NF=500,NM=3
```

The above creates a master device for set IOQUES with VSN SHARIO. This pack is shared and holds queue files; the controller type has either been previously specified for channel 2 and equipment 0, or the system default defined in CMR will be used.

```
AZ        EST    NAME=844DBL,CH=32,EQP=0,UNIT=7,ESTO=127,MOD=FREE,VSN=DB
,LDEN,SN=HICAP,MASTER=SCR,NM=2,TYPE=7154
```

The above creates a master device for set HICAP with VSN=DBLDEN. This double-density pack can be used as a scratch pack. It is at EST ordinal 127 and is connected as unit 7. The controller type is set to full track if no prior definition has been made for channel 32 and equipment 0.

```
AY          EST          NAME=844A,CH=7,EQP=5,UNIT=0,MOD=FREE
MT          EST          CH=(11,4,5),EQP=7,UNIT=0,UNITS=8D,MOD=(MTS,BID)
```

The above creates units 0 through 7 (7-track 667s) with 6684 data channel converter.

```
DS          EST          CH=10,EQP=7
NT          EST          CH=(13,12),EQP=5,UNIT=2,UNITS=4,MOD=(MTS,BID),ESTO=42
```

The above creates units 2 through 5 (9-track 669s) at EST ordinals 42 through 45.

```
CX          EST          MOD=OFF,MUX=1
```

The above creates an ECS Link using ECS buffer 1 (MUX=1) for multimainframe communication.

```
NC          EST          CH=23,MOD=OFF
```

The above creates network access device (NAD) on channel 23.

The OFF designation for RMS devices causes all record block assignment to be prevented. During deadstart, OFF drives are still checked and labeled. Only IDLE drives are ignored.

Devices in a device set having the system set attribute cannot be designated as shared (that is, the system set cannot be shared).

Parameters SN, VSN, NM, and NF are only considered at the time the device set is initialized; changing them later without reinitializing the device set has no effect. Choose the value of these parameters with a view toward future expansion.

Changing set attributes [specified by the MASTER=(SYS,PF,QUE) parameter] does not require set initialization except when the set contains a device which has been previously initialized with the attribute being added. Changing the SCR attribute does not require set initialization because it has no corresponding device attribute. The operator can change set attributes on a level 0 or 1 deadstart.

Changing device attributes [specified by the MOD=(SYS,PF,QUE) parameter] requires set initialization.

EST and/or RBR ordinals may be changed without a requirement to reinitialize the set. (All disk resident system tables are CMR independent.)

NOTE

To avoid degradation in system throughput, place the system resident device on a channel separate from any other equipment. In addition, place SYSTEM and/or PFD residency on double-ranked channels (24_8–33_8) in 6000 or CYBER 70 mainframes with more than 10 PPs or 12 channels.

Because certain tables (such as PFD, PFC, and DAM) on a device set are modified under the protection of the stack request interlock (which is issued on a controller basis), any set can interfere with the performance of other sets on the same controller. If any of the sets sharing one controller are public sets, system degradation may result.

Equipment (controller) number for 841 devices must be 4, 5, 6, or 7.

Equipment (controller) number for an 844 or 885 device must be zero; thus, a channel cannot have two 844 or 885 device controllers. No testing has been done with any other equipment on the same channel with an 841, 844, or 885.

If full-track recording mode is used for an 844 or 885 device on a 20-PP system other than a 170-class mainframe, 180-class mainframe, or a model 176 with a model D or later chassis, disk revolutions may be lost when a PP conflict occurs. The conflict occurs when the stack processor partner PP is executing I/O instructions. Although lost revolutions may occur during conflict, performance degradation is minimal.

Use an expander with single density 844-21 drives only. A 6-bit numbering scheme is used for 844-21 and 844-41 units. A site with no expander uses 00 through 07 as in the past.

With an expander on each port, use drives 00 through 07 as the first rank (first drive on each expander), 10 through 17 as the next rank, and so on. Thus, drive 35 is the fourth drive connected to the sixth port of the controller (expander five).

Designate unused disk pack entries assembled into the EST for the installation CMR as idle. This allows the operator to mount public or private devices as required.

Dual access 841/844/885 allows simultaneous data transfer to two members of a group of 841, 844, or 885 mass storage units, where:

- For 841, connect the group to two 3553-1 controllers. Each unit must have dual access option 10163 installed.

- For 844-21, connect the group to two controllers (either 7054 or 7154 types) that can be loaded with a compatible version of the controlware.

- For 844-4x, connect the group to two controllers (7054, 7154, or 7155 type) that can be loaded with a compatible version of the controlware.

- For 885, connect the group to two 7155 controllers.

There is no advantage in designating dual access if one or both of the controllers is being actively shared by two mainframes.

If the dual access configuration uses one single channel coupler, install FCO CA32618 in both controllers.

Usefulness of the dual access 841/844/885 feature, relating to the improvement of efficiency, mainly depends on the type of job mix and the number of units in the group. Even though the number of units is three, under circumstances where relatively large data transfers are expected to each of the units simultaneously or randomly, a fair improvement of throughput can be expected. Conversely, if the job mix is compute-bound and the total of RMS processing time is less than the elapsed time, or where the delay of stack requests costs nothing because of multiprogramming, no improvement is expected regardless of the number of units.

If IPARAMS symbol IP.CSET is equated to IP.C63, the data channel converters used must all be 6681s, all 6684-Is or 6684-IIs used as 6681s.

If IPARAMS symbol IP.CSET is equated to either IP.C64.1 or IP.C64.2, the data channel converters used must be all 6681s, all 6684-IIs or all 6684-Is used as 6681s.

Equipment numbers of 657 and 659 magnetic tape controllers must be 4, 5, 6, or 7. If 657 or 659 units are to be used for deadstarting, they must be configured on channels, 0, 12, 13, 32, or 33.

Equipment number for the 667 and 669 (MTS) controller must be 0. Do not use channel 0 for the MTS (66x) or ATS (67x) tape subsystems. ATS (67x) controller equipment numbers can range from 0 through 7.

Each 6000 channel can have only one 6681 or 6684 channel converter. This restriction does not exclude the use of a 6000 type controller on the same channel with the 6681 or 6684 converter.

If one of the channels is channel zero, it must appear as the first channel.

The channel must not include the high-order bit (40 octal) for the 6684; this is a function of the 6684 parameter.

6683 couplers cannot share a channel with other equipment. Use either a dedicated channel (CC) for the 6683, or turn all other equipment on the 6683 channel logically OFF.

RECORD BLOCK RESERVATION TABLE

Each mass storage device is represented by at least one entry in the RBR. Several RBRs can be generated for a single device, each describing a unique area on the device. Each entry includes a two-word header and a variable length bit table.

The first word of each RBR header contains a 6-bit allocation style code supplied as a parameter to the RBR macro when the CMR is assembled at an installation. Unique allocation style codes for each RBR can be set; this code can be used to direct a file to the RBR with a specific RB size or recording technique.

An RBR table is a single bit string of variable length, up to a maximum of 4095 bits. Each bit represents the availability of the corresponding record block (RB); the number of PRUs per RB is constant throughout the table. On a system device or on the first RBR of the master pack of a public set, the RB size can be no greater than MAXRBCNT PRUs. If the RMS deadstart feature is to be used, the system device RB size must not exceed the physical block (PB) size and should divide it exactly. The recommended RB sizes are 57 for 844 devices and 160 for 885 devices.

The RBR macro is defined in an order-independent parameter format where the parameters consist of keyword = value. Keywords and values are in the following description. All numeric values are assumed decimal unless otherwise specified. RBRs are added to CMR by inserting RBR macro statements into RBR.1 in the following form.

```
name        RBR        keyword=xxx,keyword=yyy,...
```

name	RBR name. Specify this name in the EST macro.
COUNT=rbs	Number of record blocks in this RBR (required parameter). Value must be large enough to hold disk tables in first RBR on master device. Set the counts for all the RBRs on one unit so that they do not exceed the size of the device. The number of RBs required to describe one unit of a device using standard RB size is shown in table II-1-1.
PRURB=prus	Number of physical record units per record block (RB size) with maximum limit of 4095. For an 844 device, the RB size specified must be not less than 1/32 of the PB size and not greater than 32 times the PB size. For an 885 device, the RB size specified must be not greater than 12 times the PB size. If not specified, the default value in table II-1-1 is used.
ALLOC=style	Allocation style. Value specified can be any number in the range 0 through 63. If not specified, the default value is 0.
DEFAULT=number	File assignment for files with no specified attributes. File assignment is inhibited if number is 0. Files are assigned if number is any other value or if DEFAULT=number is not specified. For more detail, refer to table II-1-2. The table shows file assignment to RBR depending on allocation style (AS) and no-attribute-file allocation bit (NAFA) values in the RBR header.
MAXRB=max	Upper threshold for DAM processing on a shared device (ignored for an unshared device). SPM returns space to the DAM when the number of locally available RBs reaches max. The default value is 1000B.
MINRB=min	Lower threshold for DAM processing on a shared device (ignored for an unshared device). SPM obtains space from the DAM when the number of locally available RBs decreases to min. The default value is 40B.

Table II-1-1. RMS Device

Device	Mnemonic	PB Size	Default RB Size (PRUs)	Count (RBs)
844-21 pack	AY	114	57	3232
844-41 pack	AZ	114	57†	3232
819 unit	AH	160	160††	4030
885 unit	AJ	320	160†	3356
.			.	
.			.	
.			.	
.			.	
.			.	
.			.	

†Requires two RBRs to fully describe disk space.
††For the 819 unit, PRU/RB cannot be changed from the default of 160. For the 819 unit, 1 PRU = 64 CM words = 1/8 sector.

Table II-1-2. Disk File Allocation

Requested File Attributes	RBR Header Values for 844-41 (AZ) Device			
	NAFA=0 AS=0	NAFA=0 AS=xx	NAFA=1 AS=0	NAFA=1 AS=xx
REQUEST(lfn,SN)	N	N	Y	Y
REQUEST(lfn,PF,SN) REQUEST(lfn,AZ,SN)	Y	N	Y	Y
REQUEST(lfn,A*xx,SN)	N	Y	N	Y
REQUEST(lfn,AZxx,SN) REQUEST(lfn,PF,A*xx,SN)	N	Y	N	Y
REQUEST(lfn,A*yy,SN)	N	N	N	N
REQUEST(lfn,AZyy,SN) REQUEST(lfn,PF,A*yy,SN)	N	N	N	N

NOTES:
- Y — Indicates file may be assigned to this RBR.
- N — Indicates file may not be assigned to this RBR.
- AS=xx — Allocation style; xx is a nonzero number.
- NAFA=n — 0 indicates no allocation; 1 indicates allow allocation.

Private devices are configured by LABELMS, independent of the RBR declarations in CMR.

Define RMS drives configured for private device usage to have no more than eight RBR table entries per device.

The 844 PB size changed from 56 to 112 PRUs/PB with the introduction of the 844 double density feature at PSR level 430. For 844 pack upward compatibility from the earlier systems to level 430 or later systems, the RB size for 844 devices is restricted to the following values. These RB sizes apply for upward compatibility regardless of whether you are using the 844 double density feature.

RB sizes less than or equal to 56 are 4, 7, 8, 14, 28, and 56

RB sizes greater than 56 may be

$(2n-1)*56+1 \le \text{RB size} \le 2n*56$ where n=1, 2,...32

Example

For n=1, $57 \le \text{RB size} \le 112$
For n=2, $169 \le \text{RB size} \le 224$

Specify consecutive sections of any device by consecutive RBR statements. You should set up the RBR table entries for drives to be used for private devices such that the maximum number of RBRs for any device is also the maximum number that will be used by any private device; and the total bit table size (controlled by the RB size and number of RB) for the drive should also be the maximum required by any private device.

The number of record blocks on the first RBR must be sufficient to hold disk tables. For a master device, the minimum number of record blocks depends on record block size, whether or not this set has a permanent file device, and the number of permanent files allowed in this set. If the number of RBs is insufficient, LABELMS will abort, stating the table that it tried to write when it ran out of space. A subsequent ADDSET may fail due to lack of space, even though LABELMS was successful. (Any size can be specified using the PRUB parameter, however, factors or multiples of 114 make better use of the storage space.)

Disk Table Space=LBL+PFT+LFT+PFD+PFC+PAM+SDT+DSR+DAM+SMT(number of RBs)

Acronym	Name	Number of RBs
LBL	Device Label	1
PFT	Physical Flaw Table	1
LFT	Logical Flaw Table	2
PFD	Permanent File Directory	NF/(4*PRURB)†
PFC	Permanent File Catalog	(NF*6)/(4*PRURB)†
PAM	PFC Allocation Map	1
SDT	Subdirectory Table	1
DSR	Deadstart Recovery RB	1
DAM	Device Allocation Map	2
SMT	Set Member Table	1

†NF is the maximum number of files allowed in the set. PRURB are the physical record units (PRU) per record block (RB).

Example for 844-21:

```
844A        RBR         COUNT=3232
```

Example for 844-41 double density disk (two RBRs are needed to fully describe disk space):

```
844F        RBR         COUNT=3232
844F        RBR         COUNT=3232
```

Example for 844-21 master device with multiple allocation styles:

```
844A12      RBR COUNT=200,PRURB=57,ALLOC=1
844A12      RBR COUNT=758,PRURB=228,ALLOC=2
```

Example of disk table space on master device:

The EST macro specifies the maximum number of permanent files (NF) as 4000. Disk table space (calculated later) will occupy 134 RBs in the first RBR.

```
PFD=NF/(4*PRURB)=4000/(4*57)=18RBs
PFC=(NF*6)/(4*PRURB)=(4000*6)/(4*57)=106 RBs

Disk Table Space = LBL+PFT+LFT+PFD+PFC+PAM+SDT+DSR+DAM+SMT
                 = 1 + 1 + 2 + 18 + 106 + 1 + 1 + 1 + 2 + 1
                 = 134RBs
```

Every RMS device is logically divided into groups of PRUs called physical blocks (PBs). The number of PBs per device must not exceed 4095.

1 PRU = 1 sector = 64 CM words

An RB represents the minimum amount of disk space that can be assigned to a file. RB size need not be equal to PB size.

If the RB size is less than the PB size but does not divide it exactly, disk space is lost. This is because an RB assignment will not be made starting in the middle of a PB if that RB would then overlap to the next PB. Instead, the RB begins on the next PB boundary and the remaining PRUs in the current PB become unavailable.

Example (PB=114 PRUs):

RB Size (PRUs)	RBs/PB	Unused PRUs/PB
14	8	2
38	3	0
56	2	2
57	2	0
110	1	4

If the RB size is greater than the PB size but is not an exact multiple of it, disk space is lost. Since an RB assignment will not be made starting in the middle of a PB and since each RB is greater than the PB size, it follows that all RBs will start on a PB boundary and that unused PRUs in the last PB of an RB become unavailable.

Example (PB=114 PRUs):

RB Size (PRUs)	PBs/RB	Unused PRUs in Last PB
169	2	59
222	2	6
228	2	0
281	3	61
1120	10	20

FLAW TABLE

The FLAW macro and the deadstart FLAW inputs have been changed to use physical addresses, in the same format as LABELMS. Numbers are assumed octal unless otherwise specified. FLAW macro entries should be inserted at FLAW.1 in the following format.

```
name        FLAW   (string)
```

Parameter	Description
name	RBR name.
string	Physical address of flaw (must be in parentheses).

The formats of the flaw strings are as follows.

Device	Format	Values	Parameters
841	Txx,Cyyy,Szz.	$0 \le xx \le 23_8$ $0 \le yyy \le 307_8$ $0 \le zz \le 15_8$	T is track number C is cylinder number S is sector number
844-21	Txx,Cyyy,Szz.	$0 \le xx \le 22_8$ $0 \le yyy \le 623_8$ $0 \le zz \le 27_8$	
844-41	Txx,Cyyy,Szz	$0 \le xx \le 22_8$ $0 \le yyy \le 1447_8$ $0 \le zz \le 27_8$	
819	Txx,Cyy,S0.	$0 \le xx \le 11_8$ $0 \le yy \le 632_8$	
885	Txx,Cyy,S0.	$0 \le xx \le 47_8$ $0 \le yy \le 1506_8$	

For all devices, specify the sector parameter (S) in the following form.

Sbbb-eee

bbb Beginning sector number
eee Ending sector number

For an 819, the sector parameter(s) need only be specified as S0. NOS/BE 1 accesses one PB (track) at a time.

Example:

```
844A          FLAW          (T12,C421,S24-26)
```

FLAW statements for the same device must be contiguous. Devices to be flawed must have SN and VSN specified in their EST macro call.

For the 844-21 (AY) and the 844-41 (AZ), NOS/BE 1 reads the utility flaw map to obtain the disk flaws. This is done when the device is a member of a public set being initialized or modified during deadstart, or when the device is labeled by LABELMS. Refer to section 17 for a description of Factory Format Support.

COMPUTING PHYSICAL ADDRESSES FOR CONVERTING RB NUMBERS

If the cylinder/track/sector of a flaw is not known, but the RB number and the PRU number are known, use the following diagrams and formulas to convert the RB and PRU to the physical address (cylinder/track/sector) of the flaw.

First, compute the PB (physical block) and SPRU (standard PRU) from the RB and PRU. If the device has one RBR and an RB size equal to the PB size, then PB = RB-1 and the SPRU is the same as the PRU. Otherwise, consult nondefault RB sizes.

819:

PB=RB

Cylinder	(PB-1)/10
Headgroup	PB-1-cylinder*10 (Headgroup is synonymous with track; resulting values should be truncated)
Sector	PRU/8 (since there are eight PRUs per 819 sector)

841:

Cylinder is PB/5.

Let

CPB (cylinder PB) be remainder of PB/5
CS (cylinder sector) be CPB*56+SPRU

Then

Track = (CS/7)mod 20
Sector = (remainder of CS/7)*2+CS/140

844-21, 844-41, and 885:

PB number

cylinder	e	m

Half track:

m = cylinder position

e = even or odd PRU number within the PB

0 for even
1 for odd

Cylinder = bits 2 through 11 of the PB number

PBS = PB size in PRUs
RBS = RB size in PRUs
TRS = track size in PRUs

TRS values are as follows.

Device	Track Size (in PRUs)
844-21	24
844-41	24
819	160
885	32

Let:

CS (cylinder sector) = PBS*m+SPRU+(integral number of SPRU/RBS)

Then:

Track = CS*2/TRS
Sector = (remainder of CS*2/TRS)*2+e

Full-track:

Cylinder = bits 2 through 11 of the PB number

Let:

EM = bits 0 and 1 of the PB number
CS = PBS*EM+SPRU+SPRU/RBS

Then:

Track = CS/TRS
Sector = remainder of CS/TRS

To compute the PB and SPRU for nondefault RB sizes, do the following:

- When the RB size is larger than PB size, one RB fits in an integral number of PBs; let this integral number be the factor.

  PB=(RB-1)*factor+PRU/(PB size)
  SPRU=remainder of PRU/(PB size)

- When the RB size is smaller than the PB size, one PB contains an integral number of RBs; let this integral number be the factor.

  PB=(RB-1)/factor
  SPRU=PRU+(remainder of RB-1/factor)*RB size

- In both cases, if the RB was not in the first RBR of the device, determine the starting PB of the RBR from the CMR assembly or dump, and add this value to the PB.

CMR EQUIPMENT CONFIGURATION EXAMPLES

Example modifications to CMR for installation of the following equipment, including 819 RMS devices (model 176 only):

415 card punch on channel 5, equipment 4

405 card reader on channel 12, equipment 4

Console on channel 10, controller 7

Two 580 PFC printers on channel 11, equipments 6 and 7, print train BCD, 64-character set

Sixteen magnetic tape units on channels 5, 11, 12, and 13 with 6681 converter, equipment number 5, units 0 through 17B

Three 9-track magnetic tape units on channel 7 with 6681 converter, equipment 7, units 0, 1, and 2

Two 844 drives (one available as a nonshared private device and one available as the system resident device and master device of the public set containing 819 devices)

Four 819 drives configured on FLPP channels 4, 5, 6, and 7 (2 by 4 access), equipment 7, units 0 through 3 (available for scratch and permanent files)

Example:

```
*INSERT      EST.1
AY           EST                NAME=844A,CH=4,EQP=0,UNIT=0,MOD=(SYS,PF,QUE),SN=PUBLSET
,,VSN=SYS000,MASTER=(SYS,PF,QUE,SCR)
CP           EST                CH=05,EQP=4
CR           EST                CH=12,EQP=4
LR           EST                CH=11,EQP=6,MOD=PFC,EC=B6
LR           EST                CH=11,EQP=7,MOD=PFC,EC=B6
DS           EST                CH=10,EQP=7
MT           EST                CH=(13,12,11,5),EQP=5,UNIT=0,UNITS=16,ESTO=10
NT           EST                CH=7,EQP=7,UNIT=0,UNITS=2
AY           EST                NAME=844B,CH=4,EQP=0,UNIT=1,MOD=(FREE,IDLE)
AH           EST                NAME=819A,CH=(4,6),EQP=7,UNIT=0,MOD=PF,SN=PUBLSET,VSN=S
,YS002
AH           EST                NAME=819B,CH=(4,6),EQP=7,UNIT=1,MOD=PF,SN=PUBLSET,VSN=S
,YS003
AH           EST                NAME=819C,CH=(4,6),EQP=7,UNIT=2,MOD=PF,SN=PUBLSET,VSN=S
,YS004
AH           EST                NAME=819D,CH=(4,6),EQP=7,UNIT=3,MOD=PF,SN=PUBLSET,VSN=S
,YS005
*INSERT      RBR.1
844A         RBR                COUNT=3232
844B         RBR                COUNT=3232
819A         RBR                COUNT=4030,PRURB=160
819B         RBR                COUNT=4030,PRURB=160
819C         RBR                COUNT=4030,PRURB=160
819D         RBR                COUNT=4030,PRURB=160
```

CMR modifications for installation of the following equipment:

415 card punch on channel 5, equipment 4

405 card reader on channel 12, equipment 4

Console on channel 10, synchronizer 7

Two 580 PFC printers on channel 11, equipments 6 and 7, print train ASCII, 64-character set

Sixteen magnetic tape units on channels 5, 11, 12 and 13, 6681 converter, equipment 5, units 0 through 17B

Three 9-track magnetic tape units on 7, 6681 converter, equipment 7, units 0, 1, 2

Five 844 drives, one available for shared private device, one available for nonshared private device, and two members of system device set designated as queue devices; channel 4, equipment 0, and units 0-4. All are accessed with a full-track controller

One 885 drive and two 844-41 drives, accessed by a 7155 controller on channels 0 and 3

Example:

```
*INSERT      EST.1
AY           EST        NAME=844A,CH=4,EQP=0,UNIT=4,MOD=(SYS,PF),SN=PUBLSET,VSN
,=SYS000,TYPE=7154
CP           EST        CH=05,EQP=4
CR           EST        CH=12,EQP=4
LR           EST        CH=11,EQP=6,MOD=PFC,EC=A6
LR           EST        CH=11,EQP=7,MOD=PFC,EC=A6
DS           EST        CH=10,EQP=7
MT           EST        CH=(13,12,11,5),EQP=5,UNIT=0,UNITS=16,ESTO=10
NT           EST        CH=7,EQP=7,UNIT=0,UNITS=2
AY           EST        NAME=844X,CH=4,EQP=0,UNIT=0,ESTO=40,MOD=(FREE,IDLE,SHAR
;)
AY           EST        NAME=844B,CH=4,EQP=0,UNIT=1,MOD=(FREE,IDLE)
AY           EST        NAME=844C,CH=4,EQP=0,UNIT=2,MOD=QUE,SN=PUBLSET,VSN=SYS0
,03
AY           EST        NAME=844D,CH=4,EQP=0,UNIT=3,MOD=QUE,SN=PUBLSET,VSN=SYS0
,04
AJ           EST        NAME=885A,CH=(0,3),EQP=0,UNIT=40,MOD=(FREE,IDLE),TYPE=7155
AJ           EST        NAME=885B,CH=(0,3),EQP=0,UNIT=41,MOD=(FREE,IDLE),TYPE=7155
AZ           EST        NAME=844E,CH=(0,3),EQP=0,UNIT=0,MOD=(FREE,IDLE),TYPE=7155
AZ           EST        NAME=844F,CH=(0,3),EQP=0,UNIT=1,MOD=(FREE,IDLE),TYPE=7155
*INSERT      RBR.1
844A         RBR        COUNT=3232
844X         RBR        COUNT=3232
844B         RBR        COUNT=3232
844C         RBR        COUNT=3232
844D         RBR        COUNT=3232
844E         RBR        COUNT=3232
844E         RBR        COUNT=3232
844F         RBR        COUNT=3232
844F         RBR        COUNT=3232
885A         RBR        COUNT=3356
885A         RBR        COUNT=3356
885B         RBR        COUNT=3356
885B         RBR        COUNT=3356
```

DEADSTART INSTALLATION PARAMETERS (DSLCOM)

IP.DSRMS defines the RMS deadstart installation parameter. If IP.DSRMS is nonzero, the code to perform a deadstart from an RMS device is assembled and replaces the code to read 60x/65x tapes for deadstart. If zero, the 60x/65x code is assembled and the RMS code is not. As a result, 60x/65x tapes cannot be used for deadstart when the system is assembled to use the RMS deadstart feature. If you must use 60x/65x tapes for deadstart but also want to use the RMS deadstart feature, a minimal recoding effort is required to replace the 66x/67x tape code with code for 60x/65x tapes. The default value for IP.DSRMS is nonzero; however, the code on the unconfigured deadstart tape was assembled with IP.DSRMS set to zero.

The release value default parameters in the Operator Option Matrix are determined by a condition micro in the deadstart parameters common deck DSLCOM. The default CMICRO OPTDF determines the initial values for the options. Deadstart option defaults can be changed by inserting, at DSLCOM.12, either a MICRO or a CMICRO. Examples to set recover I/O queues to NO and to initialize extended memory are as follows:

```
OPTDF    CMICRO    ,NNNNIN
           or
OPTDF    MICRO     1,,$NNNNIN$
```

The default of the options appear from left to right in the MICRO. To change a default value, insert a micro (named OPTDF) with the desired changes at DSLCOM.12.

The MICRO as released includes the options following (left to right in the micro string).

1.N	Reload libraries
2.Y	Recover I/O queues
3.N	Validate user sets
4.N	Equipment changes

The following two values may or may not be defined. Their assembly is governed by the IPARAMS symbols IP.ECSB and IP.SRMS.

5.U	Initialize extended memory
6.N	First mainframe to deadstart

The following value is governed by the setting of IP.ELST, except on a CYBER 180 model 990, where it is always set to Y.

7.N	Dedicated error logging

All symbols described below are defined in the common deck DSLCOM: default values are shown. A 65K memory is assumed. Central memory usage by deadstart may be modified by changing the symbol values at DSLCOM.12 on PL1A. Most symbols are keyed from a symbol defining an adjacent area, and all depend on the value of the symbol BASE. For example, if a 131K system is to have an unusually large CM resident library, it may be necessary to set the origin address of IRCP (IRADR) to a higher value by redefining BASE to any arbitrary address in the middle of CM. For central memory larger than 65K, ample space is available to enlarge both the CM resident area and the RBT area.

BASE	CEQU	112000B	Location from which origins of other areas are keyed
IRADR	CEQU	BASE-27000B	Absolute origin address of IRCP (defines size of IRCP)
MAXRBCNT	CEQU	160	Maximum RB size in PRUs of any system device or master device of a public set
TBUFO	CEQU	IRADR-2000B-(MAXRBCNT*101)	Lowest data buffer used by IRCP

CMRSIZE	CEQU	16000B	Number of words in CMR to be saved for recovery purposes
RBTSIZE	CEQU	1000B	Size of RBT area during deadstart.
DSPLCHAN	CEQU	10B	Display channel number
DSPLCTLR	CEQU	7	Display controller number
ROCKCNT	CEQU	10B	Retry count for tape parity error
DRIVBFL	CEQU	25000B	DRIVBUF length

The following dependencies and constraints exist and may be helpful in making changes.

- The central memory resident libraries/programs must not extend past IRADR or loading cannot complete. If they do, redefine BASE.

- Deadstart recovery attempts to recover the INTERCOM user tables which are located after the CM library and are typically about 5000_8 CM words. Deadstart will recover all user tables before TBUF0 and abort all users with tables above TBUF0.

- IRCP must not be larger than BASE - IRADR or it will overlay DRIVBUF. An assembly error will occur if an attempt is made to generate an IRCP larger than the current value of BASE - IRADR.

  If extended memory is to be defined (IP.ECSB nonzero), you must change the deadstart parameter IRADR. A value of BASE-32000_8 is suggested. This must be done in addition to adjusting BASE, depending upon machine central memory size.

- DRIVBUF contains copies of IRP (the RMS driver), the RMS driver overlays (one for each device type), 885 BC controlware, 844 BC controlware, and 66x BC controlware. To save CM space in deadstart, remove any of these drivers or controlware that are not needed by an installation from the deadstart tape and the size of DRIVBUF will be shortened appropriately. (Note that the controlware packages are about 3200_8 CM words each, except 885 controlware which is about 6457_8 CM words, and the 819 subsystem which is less than 1000_8 CM words.) The length of this buffer is controlled by the value of DRIVBFL.

- When the old CMR is saved for recovery, the number of words to be moved is determined by the DSLCOM symbol CMRSIZE; this value includes all the CMR tables recovered by deadstart. If IP.ECSB = 0, the JDT (job descriptor table) is the highest table recovered and CMRSIZE should correspond to the start of the extended memory parameter table (symbol T.ECSPRM in CMR). If IP.ECSB ≠ 0, the empty page stack is the highest table recovered and CMRSIZE should correspond to the start of CP.MTR in CMR; if CMR is larger, redefine CMRSIZE.

- RBTSIZE is a DSLCOM symbol which defines the maximum size of the RBT area needed for the system file. The combination of large files and nonstandard RB sizes may require you to redefine this symbol to ensure sufficient RBT space to mount public sets during post-deadstart. (Refer to Record Block Reservation Table in this section for guidelines in estimating the number of RBs needed for disk table space.)

- For a level 0 or 1 deadstart in which device labels are initialized or modified, the RMS flaws are maintained in a backward list starting at machine size - RBTSIZE. (RBTSIZE is a DSLCOM symbol which defines the maximum size of the RBT area needed for the system file.) The RMS flaws are passed to postdeadstart in a list at the end of control point zero. This list must not extend past IRADR. If it does, a warning message is displayed and the overlapping flaws are discarded.

- It is strongly recommended that the BASE origin be as high as possible, up to 377777B. Execution of IRCP is not guaranteed when loaded above 377777 minus IRCP length, nor is execution of IRCP guaranteed when the value of BASE is so high that it results in IRCP referencing buffer areas which extend above 377777B. Depending on the size of the central memory, use one of the following formulas as a guideline.

  Let:

  MAXRBT = Maximum length of RBT area expected to be used by the installation. (Deadstart recovery expects the RBT area to be intact. If the RBT delimiter cannot be found in the last 20000B CM words, recovery is not possible.)

  DOSIZE = DRIVBFL + size of OPCOM buffer (DOSIZE = 24000B on the unconfigured deadstart tape).

  BUFSIZE = Maximum of (17620B, CMRSIZE). 17620B is the combined size of buffers STLBUF, MTRBUF, SDSBUF, and CMBUF. CMRSIZE is the DSLCOM symbol used for saving CMR.

  Then, for a memory size of 131K or less, use:

  BASE = Machine size - MAXRBT - DOSIZE - BUFSIZE

  Then, for a memory size greater than 131K, use:

  BASE = 400000B - DOSIZE - BUFSIZE

Example:

Suggested values depending on the size of central memory are as follows.

Central Memory	CMRSIZE	MAXRBT	BASE
65K	16000B	10000B	124000B
98K	22000B	20000B	222000B
131K or more	22000B	20000B	323000B

- Increasing the value of BASE may restrict the capability to perform checkpoint recovery deadstarts with memory degraded.

- MAXRBCNT must not exceed 256. For machines with 65K central memory and no 885 disks set MAXRBCNT = 57 to reduce central memory reserved for disk I/O buffer space.

To help the system make the proper choices in allocating central memory between NOS/BE and NOS/VE, you can enter two parameters to indicate the minimum memory sizes that you consider necessary for your operation. For both parameters you specify the memory size divided by 1000B. The names of these parameters are:

IP.MINCM — Minimum amount of central memory to allocate to NOS/BE to help satisfy NOS/VE memory requirements. This parameter is effective only when not using UEM. Release value = 1000B.

IP.MINVE — Minimum amount of central memory needed to operate NOS/VE. Release value = 2740B.

When deadstart sets up the locations that define the various central memory allocations, it first attempts to use the standard 262K of central memory for NOS/BE. If you are using UEM, this value is fixed. When not using UEM and the amount of central memory left after subtracting 262K from the actual machine size is less than IP.MINVE, deadstart reduces NOS/BE's central memory size to IP.MINCM.

DISK PREPARATION

After generating a deadstart tape with the desired installation parameters (including IP.DSRMS set to nonzero) and the appropriate CMR configuration, perform the following steps to prepare a disk deadstart device. The CYBER Initialization Package (CIP) User's Handbook describes the procedures for setting the deadstart panel, performing coldstart, and selecting CTI options.

1. Ensure that the correct controlware is loaded and is functioning properly in all controllers that use controlware.

2. Install CIP (and possibly MSL) as explained in the CIP User's Handbook.

3. Set the deadstart panel for a level 0 deadstart from the configured deadstart tape. See the CIP User's Handbook for correct deadstart panel settings.

4. Perform a level 0 deadstart from the configured deadstart tape, making any necessary equipment configuration changes via the EST change option on the main deadstart options display (4.Y).

5. Proceed through preloading, loading, and postdeadstart. After postdeadstart completes, the message WAIT DEADSTART disappears from the left screen, indicating that preparation of the disk deadstart device is complete.

Future level 1 deadstarts can be done from the disk containing CTI. Ensure that all system devices used during this procedure are online.

To load a different system deadstart tape onto the disk without changing the version of CTI, VIVS, or MSL on the disk, perform a level 0 deadstart from the deadstart tape. If a device containing CTI is online and has the system device attribute in the EST, the system updates the disk automatically to enable future level 1 deadstarts from the disk; it is not necessary to initialize the set or device label. Refer to the CYBER Initialization Package (CIP) User's Handbook for additional deadstart procedures.

PERMANENT FILES AND DEVICE SET INSTALLATION

Under NOS/BE, all RMS devices are grouped into device sets. Each of these device sets has a setname (SN). Within the device set, the device has a unique name and a volume serial number (VSN). The setname and volume serial number are recorded in the RMS label of the device.

Device sets can be used in one of two ways: as a public set or a private set. Private sets are device sets which are typically used by a subset of jobs and, therefore, their availability may be determined by whether they are requested by any jobs. Public sets are defined and maintained by the installation and are public throughout the running of the system. The use of the public sets must be additionally qualified by the application of public set attributes to the device set. The possible attributes are the following.

- System set attribute determines all permanent files of ID=SYSTEM are to be resident within this device set. In addition, this device set contains all the system devices, including the RMS deadstart device.

- Scratch set attribute determines that scratch file assignment is to be to the applicable device sets.

- Permanent file default set attribute determines that default permanent file assignment is to be to the applicable device set.

- Queue set attribute determines that queue (input and output) file assignment is to be to the applicable device set.

The scratch set attribute is the only one which can be applied to more than one public set at the same time; otherwise, any combination of device set attributes can be applied to a public set. The maximum number of public sets is four and all attributes must be assigned. A device which is currently a member of a public set is called a public device. A device which is currently a member of a private set is a private device.

Within a device set, certain device attributes can be applied to the devices within the device set. These attributes qualify how a particular device is to be used while a member of the device set. Specifically, the device attributes include the following.

- Master device attribute defines the specified device on which the disk resident tables are to be resident.
- Permanent file device attribute defines that permanent files can be assigned to the specified device.
- Queue file device attribute defines that queue files can be assigned to the specified device. (This does not apply to private sets.)
- System device attribute defines that system files can be assigned to the specified device and that the device can contain CTI, HIVS, or MSL.

It is not possible to deadstart using 60x/65x tape drives.

The master device attribute is the only one which can be applied to only one device within the device set. If a file is not a system file and is not meant to be a queue or permanent file, it can be assigned to any device within the device set.

File assignment occurs by first picking the appropriate device within the device set. Appropriateness is partly determined by the various assigned attributes.

The device set and device attributes serve as a means of performing a software configuration of devices irrespective of the hardware configuration of the drives on which the devices are mounted. This distinction is most obvious in the case of removable devices such as 844s (pack is synonomous with device). To complete the description of the total RMS configuration, drive attributes are used. Drive attributes describe the hardware configuration of the drives. The possible attributes include the following.

- RMS type.
- Channel numbers.
- Equipment number.
- Unit number.
- Shared equipment.
- Shared unit.

Correct specification assures the accessibility of drives by the specifying mainframe.

The master device of each device set contains the following disk resident tables.

- Permanent file directory (PFD).
- Permanent file catalog (PFC).

- Set member table (SMT).
- Device allocation map (DAM).
- Logical flaw table (LFT).
- PFC allocation map (PAM).

The size of the tables is controlled by the specification of the NM and NF parameters on the EST macro call for the master device. NF declares the number of permanent files and queue files estimated to reside on the device set, and NM declares the number of devices to be members of the device set. Multiple cycles of a permanent file should be counted as one for the NF specification.

NF must be greater than or equal to 1 and less than or equal to 16384. However, if NF is greater than 15872, the number of hash points in PFD will be nonprime, which adversely affects the efficiency of the permanent file hashing algorithm.

Specification of NM affects the disk space allocated to the SMT, DAM, and LFT. The number of PRUs reserved for the DAM (disk copy of the RBR) is 2*NM. Since this number of PRUs is reserved at the time the device is initialized, you may want to add additional devices to the device set at a later time, choose NM with a view to expansion. Since generally one PRU is needed for each RBR, for multiple RBR devices NM should be increased by one for each two additional RBRs. Take special care to allow for multiple PRU DAM entries. A DAM entry uses two PRUs if the RBR bit table size is 62D CM words. For such DAM entries, increase NM by one for each additional RBR. The LFT table is the same size as the DAM table. The SMT table is assigned one record block.

The NF parameter affects disk space allocated to the PFD, PFC and PAM. The PFD is allocated NF/4 PRUs (four PFD entries per PRU). The PFC is allocated 3*NF/2 PRUs. A PFC entry always occupies an integer number of PRUs. A PFD entry has a length of 16D words.

The attached permanent file table (APF) contains two-word entries and is central-memory resident. Every permanent file in use by a job must have an APF entry. The size of the APF table (L.APF) limits the number of permanent files attached simultaneously by all jobs in the system.

The mounted set table (MST) contains five-word entries and is central-memory resident. Every mounted device set (private and public) is described in the MST. The number of MST entries (N.SETS) limits the number of device sets mounted simultaneously by all jobs in the system.

Each RMS device described in the equipment status table (EST) has an associated DDT entry in the fixed portion of the DDT (four words/entry). The remainder of the DDT is described as the variable portion of the DDT (two words/entry) and is reserved for a list of packs for which jobs are swapped out waiting.

The permanent file routines contain a universal password that, when specified in a request, grants a universal permission. A default file retention period is also defined. This universal password, permission, and file retention period apply to permanent files on private sets created on NOS/BE 1.2 level 461 and earlier systems, and on all public sets. You can change the universal password by redefining the symbol UNIV in permanent file PP routines. It is located in COMDECK PREAMB and should be defined as a nine-character value. You can change the universal permission by redefining the symbol IP.UP in deck IPARAMS on PL1A. Symbol IP.PFRP in IPARAMS defines the default file retention period. The universal password and file retention period do not apply to private sets created on NOS/BE 1.3 level 473 and later systems.

Specify a public password defined in the permanent file routines to permit use of ID=PUBLIC on a CATALOG or RENAME request. This public password applies to private sets created on NOS/BE 1.2 level 461 and earlier systems, and all public sets. You can change the public password by redefining the symbol IDPERM in permanent file PP routines. It is located in COMDECK PREAMB and should be defined as a nine-character value. This public password does not apply to private sets created on NOS/BE 1.3 level 473 and later systems.

Private sets created on NOS/BE 1.3 level 473 and later systems have their own universal password, universal permission, public password, and default permanent file retention period stored in the label of the master device. You can define default values for the ADDSET parameters UV, UP, PB, and FR by redefining symbol DFV$ (at UPWDEF.1) in the CP routine PFCCP to be nonzero and by redefining symbols UVD, UPD, PBD, and FRD (at UPWDEF.2 through UPWDEF.5) in PFCCP to contain installation defined default values. When DFV$ is set to a nonzero value, define all four default values. If you do not define any defaults, all private set master device ADDSET statements must contain UV=, UP=, PB=, and FR= parameters. If these parameters are not present, ADDSET aborts with the message xx MUST BE SPECIFIED, where xx is the parameter to be specified.

You must provide a PP routine (part of your own accounting routines), to store the user's account number into each control point area, in word W.CPFACT. This account number is presumed to have been taken from the job statement, and is used for CATALOG regardless of any AC parameter specified in the control statement or macro call. The identification, one to nine alphanumeric, display-coded characters, has the following format.

59 5	0
Account No. (Right Justified)	16

The account number should be right-justified to the 16 code (octal) and left-filled with binary zeros. When W.CPFACT is nonzero, accounting dayfile messages are issued to both system and control point dayfiles whenever the status of a permanent file changes; that is, when a catalog, purge, or rename is processed.

DISK SPACE THRESHOLD SETTING

The unavailability of certain types of disk space can cause deadlocks. These types and the kinds of files that reside on them are listed.

Public Set	Device Attribute	Kind of File
System	PF,SYS	Dayfile, CE error file
PF	PF	Default PFs
Q	Q	Default Q files
SCR (can be multiply defined)	--	Default local files, swap/roll files

Two thresholds for each of the four types of disk space are specified as assembly parameters in 2RN. 1RN periodically (every 2**IP.RBINT seconds) initiates the calculation of the sum of available RBs for each of the thresholds. If the available disk space equals or falls below the first or upper threshold, an operator warning message is issued. If the available space falls below the second threshold, the system is placed in step mode and a final operator warning message is issued. The upper threshold for scratch space should be designated high enough (CM size plus 10 percent) to enable the operator to clear control points and initiate one or more jobs to free space.

The system issues only one message for each threshold. When available disk space again exceeds a threshold, a message informs the operator. Disable the feature by defining IP.RBINT=11D.

The interval at which the available RBs are checked against the threshold values is defined in 1RN as an installation assembly parameter (IP.RBINT). The interval in seconds is calculated as 2**IP.RBINT. The default value of IP.RBINT equals two (that is, a 4-second interval).

Macro DSTDEF generates the disk space threshold table (DSTT) used by 2RN to calculate the available RBs on the sets defined in the DSTT. The set being checked must have the attributes defined in the macro; however, it may have other attributes as well. This allows installations which have sets with multiple set attributes (for example, PF and Q) which in turn contain devices with a single attribute (for example, a PF device or a Q device) to check the set once for available PF space and once for available Q space with independent thresholds.

On a shared disk, all local space will be released by any mainframes having the device logically OFF in the EST. This may prevent space deadlock situations.

DSTDEF has the following format.

DSTDEF	T1,T2,A,B,C,D
T1=	Upper threshold (in RBs)
T2=	Lower threshold (in RBs)
A through D	Optional parameters indicating up to four set attributes required on sets to which the thresholds are applied

P	Permanent files
Q	Queue
S	System
X	Scratch

Insert calls to the DSTDEF macro at DSTDEF.1 in deck 1RN on PL1A.

Default threshold settings are as follows.

```
DSTDEF        T1=50,T2=20,A=S
DSTDEF        T1=100,T2=50,A=P
DSTDEF        T1=150,T2=100,A=Q
DSTDEF        T1=160,T2=80,A=X
```

Threshold examples:

Configuration 1

Setname	Set Attributes	VSN	Device Attributes
SYSTEM	SX	844A	MSP
PFQSET	PQ	844B	MPQ
PFQSET	PQ	844C	PQ
PFQSET	PQ	844D	PQ

DSTDEF macro calls

DSTDEF	T1=200,T2=100,A=S,B=X
DSTDEF	T1=300,T2=150,A=P,B=Q

In this example, the operator is warned when the available RBs on VSN 844A reaches 200 or fewer and 100 or fewer, and when the available RBs on VSNs 844B through 844D combined reach 300 or fewer and 150 or fewer.

Configuration 2

Setname	Set Attributes	VSN	Device Attributes
SYSTEM	SX	844A	MSP
PFQSET	PQX	844X	MP
PFQSET	PQX	844Y	Q
PFQSET	PQX	844Z	P

DSTDEF macro calls

DSTDEF	T1=250,T2=150,A=X
DSTDEF	T1=100,T2=50,A=S
DSTDEF	T1=200,T2=100,A=Q
DSTDEF	T1=150,T2=50,A=P

In this example, the operator is warned when the available space for scratch files (VSNs 844W, 844X, 844Y, and 844Z) is 250 and 150 RBs (or fewer). Warnings for PF space occur when VSNs 844X and 844Z contain 150 and 50 available RBs (or fewer), for Q space when VSN 844Y contains 200 and 100 available RBs (or fewer), and for S space when VSN 844W contains 100 and 50 available RBs (or fewer).

SCHEDULING PARAMETERS

JOB CLASSES

Definitions:

- Minimum queue priority (MINQP); the priority with which a job will first enter the CM queue.
- Maximum queue priority (MAXQP); the maximum priority level a job in the CM queue may achieve while waiting for scheduling.
- Base quantum (BQ); the amount of time that a job, once brought to a control point, maintains a high priority, thus helping to avoid being swapped out by another job.
- Quantum priority (QP); the priority given to a job when it has been swapped in. The job maintains that priority for the duration of its base quantum.
- Age rate (AR); a factor used to weight the priority of a job according to the time spent in the CM queue.
- Anticipated field length (AFL); an amount of central memory field length which the scheduler tries to set aside in anticipation of jobs of INTERCOM or higher class. Scheduler will not swap in a job from the central memory queue if it is a batch or device class job and if such a swap would not leave at least an amount of unassigned memory equal to AFL.

The first five parameters apply to each of the available classes of jobs. Each class serves to define a series of jobs by their common characteristics, such as response time requirements or the minimum amount of time that a job has access to core.

The seven classes are:

Batch

Device (batch with nonallocatable devices)

INTERCOM

Multiuser

Express

Graphics

Extended memory (batch with direct access extended memory)

When a job requests scheduling for central memory, its job descriptor table entry is placed into the central memory queue with a queue priority equal to the minimum queue priority of its class. Its priority is evaluated according to its minimum queue priority, the age rate of the class, time in the queue, and the job statement priority. When the priority of the job reaches the maximum for the class, aging ceases. This priority evaluation is performed for all jobs in the central memory queue, and the results are compared with the priorities of those jobs at control points. When a job is swapped into central memory, it is given a priority equal to the quantum priority of its class.

When the quantum of the job has elapsed, its priority is reduced to the minimum of its class.

Since it requires some overhead to swap a job, the quantum permits a job to remain at a control point for a reasonable length of time before it becomes eligible for swapping. The quantum of a job is considered elapsed when the job has used a specified amount of CPU or PPU time.

All priorities for a class, except MAXQP, are weighted by job statement priority.

Figure II-1-2 illustrates the interaction between two classes, batch and INTERCOM, and displays in a graphical form the relationship between the parameters of these two classes.

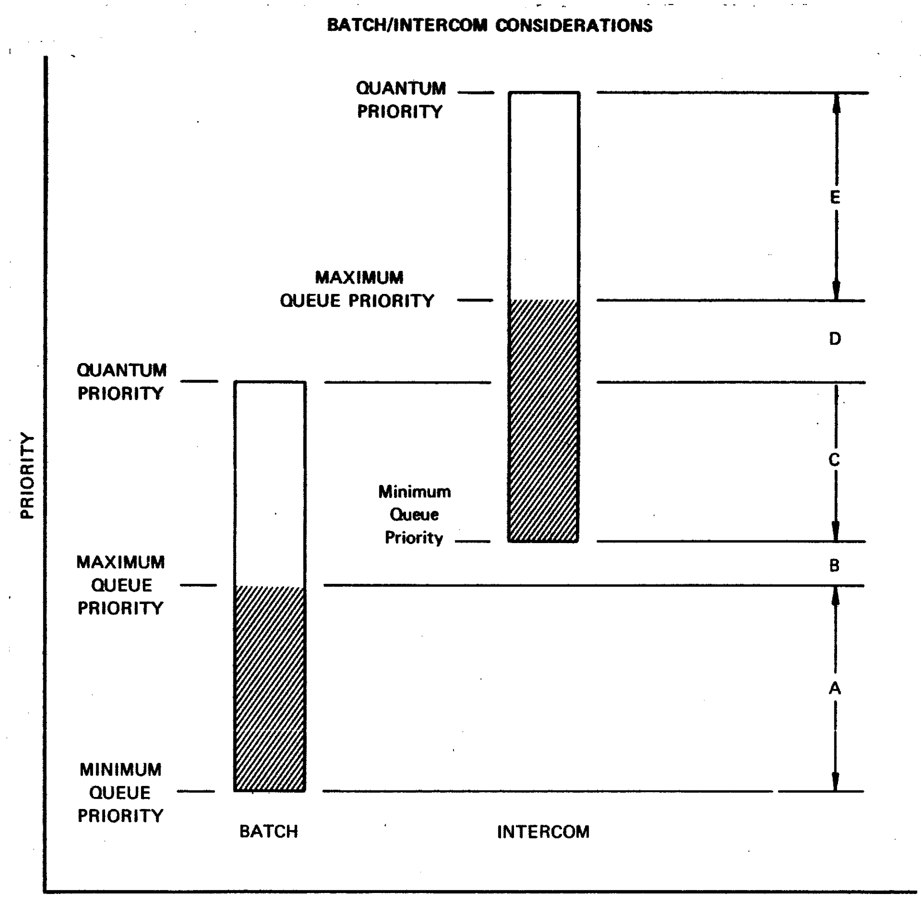


Figure II-1-2. Batch/INTERCOM Considerations

The assumptions used in formulating this set of parameters were that the response time for INTERCOM users should fall within certain bounds, irrespective of the batch loading; that, once a batch job is executing, it has a guaranteed period of execution before competing with other batch jobs; and that a batch job be allowed to execute a minimum period before a swapout can be forced by an INTERCOM job. Within the batch class, the aging between the minimum queue priority and maximum queue priority (interval A) is intended to ensure that, job statement priority considerations aside, the first batch job to enter the central memory queue will be the first job to be swapped into central memory. The minimum queue priority for an INTERCOM job is greater than the maximum queue priority of the batch job (interval B) so that INTERCOM jobs will not have to compete with batch jobs waiting for central memory. Aging of jobs in the INTERCOM class serves two purposes: first, as in the batch class, to ensure first into the central memory queue, first into central memory; secondly, to allow INTERCOM jobs, after a certain period of time has elapsed, to force the swapout of a batch job so that the INTERCOM job can run.

The extra increment D, between the quantum priority of a batch job and the maximum queue priority of an INTERCOM job, allows INTERCOM jobs to be selective in the batch jobs that are swapped out to provide core, by becoming eligible to swap, first of all, low job statement priority batch jobs and, eventually, to be able to force out even the highest job statement priority batch jobs. Interval D can be set smaller than the total range of the job statememt priority values. By doing so, those jobs with a high job statement priority will not be forced out by INTERCOM jobs before their quantum has expired. Interval E between the maximum queue priority and the quantum priority of INTERCOM jobs, is used for similar purposes as interval C in the relationship between INTERCOM jobs and the next higher class of users. Similarly, this also will allow INTERCOM jobs to run to their quantum before they start to compete for central memory with other INTERCOM jobs.

The following list is the default set of parameters, as they appear in CMR. The parameters selected provide good throughput for an installation running a heavy load of batch jobs, as well as provide good response time for a 20-terminal INTERCOM system where an average of ten terminals are active at any one time. However, graphics jobs, if running, will take precedence over all other jobs.

```
*                 SCHEDULER PARAMETER SETTINGS
*
*                                         PARAMETER             $DISPLAY
*                                         DESCRIPTION           TYPEIN
*MAXNBA     CEQU       30B          MAX NO OF JOBS W/O        MAXN1
*                                   NON-ALLOC EQUIP
MAXNDE      CEQU       10B          MAX NO OF JOBS WITH       MAXN2
*                                   NON-ALLOC EQUIP
MAXECC      CEQU       4000B        MAXIMUM ECS               MAXE
                                    COMMITMENT
QPINP       CEQU       2200B        INPUT QUEUE QUANTUM       QP0
*                                   PRIORITY
BQINP       CEQU       2000B        INPUT QUEUE BASE          BQ0
*                                   QUANTUM
BONECS      CEQU       1            DAECS PRIORITY BONUS      BON
*
*                      BATCH CLASS
*
MINQPBA     CEQU       100B         MIN QUEUE PRIORITY        MINQP1
MAXQPBA     CEQU       1000B        MAX QUEUE PRIORITY        MAXQP1
ARBA        CEQU       4B           AGING RATE                AR1
QPBA        CEQU       1400B        QUANTUM PRIORITY          QP1
BQBA        CEQU       2000B        BASE QUANTUM              BQ1
*
```

```
*                          DEVICE CLASS
*
MINQPDE        CEQU        200B        MIN QUEUE PRIORITY          MINQP2
MAXQPDE        CEQU        1000B       MAX QUEUE PRIORITY          MAXQP2
ARDE           CEQU        10B         AGING RATE                  AR2
QPDE           CEQU        1400B       QUANTUM PRIORITY            QP2
BQDE           CEQU        2000B       BASE QUANTUM                BQ2
*
*                          INTERCOM CLASS
*
MINQPIN        CEQU        1110B       MIN QUEUE PRIORITY          MINQP3
MAXQPIN        CEQU        2400B       MAX QUEUE PRIORITY          MAXQP3
ARIN           CEQU        1000B       AGING RATE                  AR3
QPIN           CEQU        2500B       QUANTUM PRIORITY            QP3
BQIN           CEQU        200B        BASE QUANTUM                BQ3
*
*                          MULTI-USER CLASS
*
MINQPMUJ       CEQU        2410B       MIN QUEUE PRIORITY          MINQP4
MAXQPMUJ       CEQU        2510B       MAX QUEUE PRIORITY          MAXQP4
ARMUJ          CEQU        200B        AGING RATE                  AR4
QPMUJ          CEQU        3000B       QUANTUM PRIORITY            QP4
BQMUJ          CEQU        4000B       BASE QUANTUM                BQ4
*
*                          EXPRESS CLASS
*
MINQPEXP       CEQU        1000B       MIN QUEUE PRIORITY          MINQP5
MAXQPEXP       CEQU        3200B       MAX QUEUE PRIORITY          MAXQP5
AREXP          CEQU        400B        AGING RATE                  AR5
QPEXP          CEQU        3200B       QUANTUM PRIORITY            QP5
BQEXP          CEQU        400B        BASE QUANTUM                BQ5
*
*              GRAPHICS CLASS
*
MINQPGRA       CEQU        2000B       MIN QUEUE PRIORITY          MINQP6
MAXQPGRA       CEQU        3500B       MAX QUEUE PRIORITY          MAXQP6
ARGRA          CEQU        2000B       AGING RATE                  AR6
QPGRA          CEQU        3500B       QUANTUM PRIORITY            QP6
BQGRA          CEQU        1000B       BASE QUANTUM                BQ6
****
               EXTENDED MEMORY CLASS
MINQPECS       CEQU        200B        MIN QUEUE PRIORITY          MINQP7
MAXQPECS       CEQU        770B        MAX QUEUE PRIORITY          MAXQP7
ARECS          CEQU        4           AGING RATE                  AR7
QPECS          CEQU        2000B       QUANTUM PRIORITY            QP7
BQECS          CEQU        6000B       BASE QUANTUM                BQ7
```

NOTE

The values for QPxx, MAXQPxx, and MINQPxx must not exceed 6777B. The system uses priorities greater than 6777B to indicate special conditions.

The foregoing default set of Scheduler parameter settings does not represent ideal settings for most installations. Adjustments should be made to these parameters so as to match the needs of individual sites. Although the default parameters produce reasonable batch throughput and interactive job response time on a 131K mainframe, the parameter set must be modified for efficient use on a smaller mainframe. The following settings are recommended as a starting point for further tuning systems to individual site requirements.

The 131K parameter set should appear similar to the following.

	IP.MPR	70B
	MAXNBA	6
	MAXNDE	2
	QPINP	2000B
	BQINP	2000B
BATCH CLASS	MINQPBA	0
	MAXQPBA	700B
	ARBA	1
	QPBA	1000B
	BQBA	2000B
DEVICE CLASS	MINQPDE	0
	MAXQPDE	700B
	ARDE	1
	QPDE	1000B
	BQDE	2000B
INTERCOM CLASS	MINQPIN	700B
	MAXQPIN	1420B
	ARIN	100B
	QPIN	1520B
	BQIN	200B
MULTI-USER CLASS	MINQPMUJ	1420J
	MAXQPMUF	1520J
	ARMUJ	200B
	QPMUJ	1620B
	BQMUJ	7000B
EXPRESS CLASS	MINQPEXP	400B
	MAXQPEXP	1420B
	AREXP	200B
	QPEXP	1420B
	BQEXP	4000B
GRAPHICS CLASS	MINQPGRA	1420B
	MAXQPGRA	1530B
	ARGRA	200B
	QPGRA	1630B
	BQGRA	2000B
EXTENDED MEMORY CLASS	MINQPECS	0
	MAXQPECS	670B
	ARECS	1
	QPECS	1400B
	BQECS	6000B

The parameter set for a 65K mainframe should be identical with the exception of MAXNBA dropping to a value of 2.

The batch classes have low minimum and maximum queue priorities as well as low age rates. The device class with twice the age rate of the batch class gives the device class a scheduling advantage over the batch class. A device class job would experience, on the average, half the wait time of a batch class job. Since the device class represents additional resources being tied up, such as control points and tapes, it is preferable to get that job through the system with a minimal delay.

The quantum priorities of the batch and device classes are low enough so that INTERCOM Jobs, having a high age rate, can force batch jobs to be swapped out after a one-half to two second delay, depending on job card priority and quantum considerations.

The INTERCOM class job is given a small base quantum which normally is enough time to execute an INTERCOM job step. The batch quantum, on the other hand, is larger, preventing batch jobs from swapping other batch jobs unnecessarily. The multiuser class job, such as EDITOR, is given the highest priority because it can service several INTERCOM users simultaneously.

The parameter MAXN determines the maximum number of batch or device class jobs which can run at any given time. The number of device class jobs is kept small; the determining factor being that device class jobs are rolled out rather than swapped out; each job can make a control point unavailable for swapping. It is essential to keep a reasonable number of control points available for serving other jobs. The maximum number of batch jobs is much higher, a large number being preferable to provide the Scheduler with a better pool of job candidates, allowing better core utilization. However, too large a job pool may adversely affect individual job turn-around while improving total system throughput.

Parameters QP0 and BQ0 in the lower half of the S display are the quantum priority and base quantum given to jobs coming out of the input queue and entering a control point for the first time. The quantum priority is higher than that for normal batch jobs, enabling short jobs to run to completion without swapping.

The express queue is given a high priority and aging rate, since it contains all jobs terminated by operator intervention. The quantum is small because the end-of-job procedure is minimal. This class was given express consideration under the assumption that these jobs would release valuable resources back to the system.

CENTRAL MEMORY

The determination as to which jobs may occupy central memory simultaneously depends on the job class, job field length, and the available central memory field length. Additional considerations affect central memory allocations such as maximum field length (MFL) and anticipated field length (AFL).

MFL This value represents the largest amount of central memory field length any single job may obtain. MFL is set by installation parameter IP.MFL but may be changed by the protected DSD S-display command, MFL,nnnn, which changes MFL to nnnn * 100B CM words. (nnnn must not exceed 3777B.)

AFL This value is the CM FL which Scheduler will set aside for jobs of INTERCOM or higher classes. Only batch or device class jobs waiting in the CM queue are affected by this parameter as the value of AFL is subtracted from the field length available to these two job classes before determining if sufficient field length is available for the job. AFL is initially set to AFL.BAS * 1000B CM words. When INTERCOM is up, AFL is incremented by AFL.INT * 1000B CM words. As with MFL, AFL may be changed by the protected DSD S-display command, AFL,nnn, which changes AFL to nnn * 1000B CM words.

AFL is used to minimize the effect of the following events.

When an INTERCOM Job step ends and the control point and field length are freed, Scheduler assigns this control point and field length to a batch job from the CM queue. Another INTERCOM job may displace this or another batch job because INTERCOM jobs have a higher priority. When this happens frequently, the repetition of this cycle can become a real problem.

To reduce the frequency of this swapping, Scheduler requires that a certain amount of CM remain unassigned and thus available for initiation of an INTERCOM job step, but not available for swap-in consideration to batch or device class jobs. This amount of CM is called the anticipated field length (AFL). The value AFL is defined by two symbols in CMR.

AFL.BAS Basic AFL is used if INTERCOM is not up.

AFL.INT INTERCOM increment is the amount added to basic AFL when INTERCOM is brought up and subtracted when INTERCOM is dropped.

Use the following guidelines to arrive at a practical setting for MFL and AFL for each individual installation.

PFL Physical Field Length, in CM words

CMR Size of CMR (LWA of library)

RBT Size of RBT area

POSFL 2 * IP.POSFL * 100B

FFL Fixed field length, total field length in use by system routines (JANUS, INTERCOM, SCPs, 7000 Station, and so on)

UFL User field length, available for user jobs

MUFL Maximum UFL (equal to UFL when FFL=0) MUFL = PFL - CMR - RBT - POSFL

AFL.INT should be set to a value which approximates the average FL occupied by INTERCOM jobs at one time.

If your installation experiences a large number of swapouts caused by requests for increased FL by jobs at control points, increase AFL.BAS to counteract this effect.

You may set MFL to any appropriate value within the following limitations.

- Too small a value may preclude some standard software products from executing.
- If MFL is set greater than MUFL, a job requesting this much FL will be swapped out and cannot be swapped back in to complete.

If MUFL>MFL>UFL, a job requesting a field length of MFL will be swapped out and cannot be swapped back in to complete until at least one of the system routines is terminated. This may occur if a system routine is initiated during the time that a very large job is executing. This condition may effectively remove nonallocatable resources such as tape drives or permanent files from the system when, for example, INTERCOM is initiated. Advise console operators that this problem may occur.

If at any time MFL UFL - AFL, a job in batch or device class which has (or is requesting) a field length of MFL may be swapped out and Scheduler will not normally swap it back in. The job may be allowed to swap back in by reducing AFL or by dropping INTERCOM, or by locking the job in with the N.LOCKIN. command. Another alternative is to drop a system routine.

It is not recommended that MFL be changed dynamically while the system is processing jobs. Change it only during system initialization to correct an incorrect setting of the parameter when the system was built. Otherwise, certain jobs abort when they request additional memory.

EXTENDED MEMORY USE

Extended memory use is affected by several factors that can be changed by console entries.

MFLE — The largest amount of extended memory that any user can obtain. This value is established by IP.MECS. Refer to the NOS/BE Operator's Guide for a description of the MFLE S-display command.

MAXECS — The total amount of direct access extended memory committed to active jobs. For efficient use of extended memory, this value should normally be set higher than the actual amount of available extended memory. Too high a value will cause excessive scheduling and swapping. Refer to the NOS/BE Operator's Guide for a description of the MAXE S-display command.

BONECS — The amount of extra priority given every job with extended memory assigned. System efficiency improves if the swapping of user field lengths can be minimized. Refer to the NOS/BE Operator's Guide for a description of the BON S-display command.

CPU

CPMTR selects which job the CPU should be assigned to next. Using the parameter values with which the system is released, this selection occurs at least every 20 milliseconds for each CPU.

The selection is based on the CPU priority level associated with each job. As released, there are five possible levels. They are defined by the following symbols which are defined in IPARAMS.

PR.IDLE (0)	A low priority job for default assignment.
PR.BATCH (1)	All normal batch jobs execute at this priority.
PR.INT (PR.BATCH+1)	INTERCOM job.
PR.SCP (PR.INT+1)	System control points and jobs initiated by the operator.
PR.SYS (PR.SCP+1)	System jobs (storage move and scheduler).

You may redefine each of these levels by inserting a new value into IPARAMS in the same manner that IP.xx symbols are redefined. Note that PR.INT is defined relative to PR.BATCH, and so forth, so that when any one level is redefined, all higher levels are automatically redefined. PR.SYS must always be the highest priority level.

Because INTERCOM jobs have a higher CPU priority, it is possible for an errant INTERCOM job to seriously degrade the batch jobs. To avoid this possibility, insert the statement PR.INT CEQU 1 into IPARAMS. This leaves INTERCOM jobs running on an equal basis with BATCH jobs.

SYSTEM IDLE MODE

If IP.SIDLE is nonzero, code is assembled to support an IDLE mode of operation. When the system is in IDLE mode, control point activity is inhibited; the CPU is not scheduled to any jobs at control points and no jobs are initiated or swapped into vacant control points. If time permits, IDLE mode swaps out all control points and performs a system level checkpoint. Such a checkpoint may be used during deadstart to recover the system even if memory contents are subsequently lost.

The system uses IDLE mode to control activity and possibly improve system recoverability in the event of circumstances which jeopardize system availability. IDLE mode may also be initiated by operator command; in this case, a checkpoint is always performed. This feature may be useful in idling system activity prior to a scheduled system downtime (for example, prior to a preventive maintenance period).

NOTE

IDLE mode does not provide a completely restartable system checkpoint. Jobs which cannot be swapped out are not recovered but are either rerun or dropped if they have no-rerun status. Included in this class are jobs with nonallocatable equipment assigned, jobs waiting for operator action, and real-time jobs.

EXTENDED MEMORY INSTALLATION PARAMETERS (ECSCOM)

The extended memory extensions are designed primarily to improve the efficiency of an I/O bound system by accomplishing the following:

- Buffering the sequentially accessed RMS files through extended memory.
- Swapping jobs to extended memory.
- Moving a part of the system library to extended memory.
- Allocating files in extended memory.
- Moving part of CMR code to extended memory to free up CM space.

Define the default values of ECSCOM configuration parameters with the CEQU or CMICRO macros, so that you can insert all modifications at one given place. Use the CEQU and CMICRO macros to define variables conditionally. Since they are effective only if the variables have not been previously defined, any modifications should precede the default definitions.

Define installation parameters oriented to extended memory in the COMDECK ECSCOM. Changes may be made at ECSCOM.8 in an update of PL1A. Default values and parameter descriptions:

IP.EBUF (16D)

Defines the default extended memory buffer size in pages. To significantly improve system I/O, the extended memory buffer allocated to a file should be at least four times larger than the buffer used in CM for the same file, resulting in a default value in the 10000 to 20000 (octal) words range. A larger extended memory buffer (40K or more) does not provide any significant improvement compared with the default value.

If an extended memory buffered file does not overflow its buffer, it stays in extended memory and is processed as an extended memory resident file, possibly locking a very large amount of extended memory for only one file. Buffer space is not reserved when the buffer is requested; it is allocated only when needed and released as soon as possible, one page at a time. Allocation of an extended memory buffer to a file having a CM buffer approximating one RB does not improve throughput because of the scheduling algorithm used by the stack processor.

If 819 buffering is enabled (IP.819 is nonzero), the parameter IP.EBUF has no effect for 819 buffering. The default buffer size is not variable.

IP.ELIB (0)

If zero, the code for extended memory resident library will not be assembled in the system; if nonzero, maximum number of words/1000 (octal) that may be used for storing extended memory resident library programs. This value can be changed at deadstart time; however, it can be nonzero only if IP.ECSB is nonzero. IP.ELIB cannot be set equal to the paged extended memory total length, since part of it is issued for the system area including empty page stack, system subpages (at least 2K), preallocation page reservation table (1K), and extended memory system buffers.

IP.ERES (0)

If set to one, the extended memory resident file capability is activated. The extended memory resident file option feature can improve system throughput for a given job by keeping large files (particularly random access) in extended memory. However, it can have an adverse effect on the overall improvement of the system by drastically reducing the amount of extended memory available for job swapping, extended memory buffering, and the system library.

IP.ECSW (0)

If IP.ECSW is zero, the code that allows swapping of user direct access extended memory is disabled. If not zero, batch jobs with extended memory requirements (but with no nonallocatable device requirements) will be assigned to the extended memory class, and will be scheduled in the same way as batch class jobs. Note that more resources will be used to swap extended memory field lengths to mass storage, but system throughput will improve. If IP.ECSW is nonzero, IP.ECSB must also be nonzero.

SYSTEM CIRCULAR BUFFERS (SCB)

Data is buffered between extended memory I/O buffers and disks by the system circular buffers. There are two types of SCB: CM and DDP. In both cases, the data transfer is controlled by the 1SP I/O buffering executive and circular buffer manager (CBM), a CP monitor function. CBM is activated by 1SP to start and to end the transfer and by MTR during the transfer through a system M.BUFFER function. The transfer is controlled by the SCB control table which has FIRST, IN, OUT, LIMIT, and TRIGGER pointers. The CM circular buffer that the control table points to is 101B words/PRU in length for a CM SCB and 1 word/PRU in length for an DDP SCB. Minimum size for an SCB is 3 PRUs. Larger SCBs cause fewer CPU interrupts, but require more extended memory to be allocated for a read ahead.

SCBs are defined by the ECSBUF macro as follows:

```
ECSBUF        size,DDP

   size   PRU count; SCB size

   DDP    blank; CM SCB
          DDP; DDP SCB
```

There must be one ECSBUF macro call for each SCB defined. In addition, each DDP SCB must have a DDP defined in EST. If more DDPs are defined in the EST than there are DDP SCBs, the additional DDPs are not used. If fewer DDPs are defined in the EST than there are DDP SCBs, the additional SCBs are not used. The number of CM SCBs should be limited to three. The number of DDP SCBs corresponds with the number of available DDP channels. Also, the number of SCBs must be the same as the number of RMS controllers.

An example of SCB installation changes follows.

Assuming that extended memory buffering is defined (IP.ECSB = 1), there is one DDP available and if two CM SCBs are to be defined, the installation changes are as follows:

```
*INSERT IPARAMS.15
(other installation parameter modifications)
IP.ECSB CEQU1
*INSERT EST.1
(other EST modifications)
ED    EST    CH=12,EQP=5
*INSERT    ECSBUF.1
           ECSBUF   24,DDP
           ECSBUF   16
           ECSBUF   16
```

Data can be transferred between extended memory and RMS using either type of SCM interchangeably. The priority of the SCBs, as determined by the order of the ECSBUF macros, determines which SCB is assigned when one is requested. The first SCB that is not busy is the one assigned. In the previous configuration, three SCBs are defined with DDP having the highest priority.

At least one SCB is required if extended memory swapping (IP.ECSW) is enabled.

MACROS TO CONSTRUCT EXTENDED MEMORY LABEL

Before extended memory is used for the system,† it is divided into partitions and a label that defines these partitions is written to extended memory. The image of an extended memory label can be set up in CMR for deadstart processing, so that the operator can construct an extended memory label when required.

An extended memory label consists of a two-word header and two-word entries for each partition. The number of partitions is limited to nine (one for a COMMON partition and two per mainframe for as many as four mainframes).

The extended memory label header contains the length of defined extended memory and the number of defined partitions with a specially formatted header word and a checksum. The extended memory label is written in the first 1000B words of extended memory to one of the areas starting at 120B, 230B, 340B, 450B, 560B, or 670B.

Two macros are available to set up a temporary image of an extended memory label in CMR at location 340B. This information will be initialized during deadstart continuation.

```
ECSLABEL    length
```

length — Defined extended memory length/1000B. On systems for 180-class mainframes, the extended memory field length is the UEM field length minus the storage needed for EI and machine page tables (approximately 20000B words). On 180-class mainframes that have more than 16 megabytes (10000000B words) of memory, the UEM field length cannot be greater than 1,835,008 (7000000B) words. The maximum length that can be specified on the ECSLABEL macro is 7777B.

†When constructing the extended memory label, extended memory on a model 176 or 180-class mainframe cannot be shared between mainframes; all partitions of extended memory must be associated with the model 176 or 180-class mainframe.

Two macros are available to set up a temporary image of an extended memory label in CMR at location 340B. This information will be initialized during deadstart continuation.

```
ECSLABEL        length
```

length — Defined extended memory length/1000B. On CYBER 8x5 systems, the extended memory field length is the UEM field length minus the storage needed for EI and page tables.

```
ECSPART name,type,fl,bit,fw
```

name — Partition (up to 10 characters), which is the mainframe ID unless the partition is type COMMON.

type — Partition type:

1 Direct access extended memory area. Assign one partition of this type to each mainframe ID. This area contains the extended memory system segments.

2 System area and paged extended memory area. Assign one partition of this type to each mainframe ID. This area includes extended memory system tables and buffers and may contain extended memory resident libraries and files.

3 COMMON area for multicomputers. First partition defined in extended memory must be type COMMON if ECS link is defined.

fl — Partition length/1000B. Total partitioned field length should be at least 1K less than specified on the ECSLABEL card to accommodate the ECS label itself. Maximum partition length is 4000000B.

bit — Reserved for future use.

fw — First word address/1000B of partition. If absent, LWA+1/1000B of preceding partition is assumed.

The following example assumes the size of extended memory is 754000B (250K decimal) words and divides extended memory for two computers and a COMMON partition.

```
ECSLABEL    754
ECSPART     LINK,3,10
ECSPART     ONEOVEM,1,40
ECSPART     THEOTHER,1,40
ECSPART     ONEOVEM,2,300
ECSPART     THEOTHER,2,300
```

43000B words of ECS remain unassigned in the example.

The definition of ECS label can be placed at CMR.2167.

The computer identification label for a mainframe may optionally be stored in CMR. When partitions are assigned at deadstart, the mainframe ID from CMR is compared against ID of the partitions. The operator can modify the mainframe ID from the console during deadstart extended memory partitioning processing.

A computer ID is stored into CMR by defining the configuration parameter for computer ID, IP.CMPID. Place this definition, if changed, at CMRIP.1.

Example: IP.CMPID   CMICRO   10,(SN58)

ECS SYSTEM SEGMENTS

The segments in an ECS system have all their nonlocal symbols defined in text CMRTEXT. If none of those symbols are affected, the segments need not be reassembled when configuring a system. Deck CMRDIR defines the residence of the segments. A segment must be named in an OVL macro call.

The format of the call follows:

```
OVL     segment, CM, iparam
```

segment — The segment name

CM — Indicates segment should be resident, no matter where it is defined.

iparam — Installation parameter when specified. The segment is only defined if the installation parameter is nonzero.

OVL calls follow a call to the AREA macro, defining the overlay area where the corresponding segments execute.

```
AREA name
```

Three areas are defined for use as name.

CM — Central memory resident.

MTR — Monitor mode overlay area.

USER — User mode overlay area.

You may move segments to CM residence by defining them within the CM resident area, or by adding the CM parameter to the corresponding OVL call. No other change of area is allowed.

CMRDIR as released contains a complete template for the system. All segments defined in the CM area are resident segments and cannot be moved to another area. Do not make changes to CMRDIR unless they result in significant performance improvement. Segments not defined in CMRDIR are not included in the ECS system built by LDCMR.

SEGMENT ACTIVITY COUNT

A provision has been made to count how many times each ECS resident segment is being referenced to aid the installation in determining how many segments should be moved into central memory. This count may be activated via an SAC call in RA+1. The interface description follows. You can write a user program to request the data and format it into a usable report.

CALLING SEQUENCE

```
label   SYSTEM,SAC,R,addr
```

R is required recall parameter.

The area specified by addr is at least twice the number of segments in length.

The code for counting segment calls replaces the code that performs the segment trace. The counts are accumulated in the segment trace buffer. On each call, the counts are transferred from the trace buffer to the user's buffer at addr, and the trace buffer is reset to zeros.

The first time that the segment activity count is called, the trace buffer contains trace data instead of segment activity counts. In this case, the data that is returned should be ignored.

The format of the data returned at addr follows.

addr contains the length of the data that was returned. addr/2 is the number of segments.

addr+1 is zero.

addr+2n is the name of segment number n, left-justified, zero-filled.

addr+2n+1 is the number of segment calls since the last call for segment activity counts.

The DSD command, RESTART, causes tracing to be resumed.

If ADDR+2n is not within the field length of the calling program, the job is aborted with the message PP CALL ERROR.

If the trace buffer is smaller than n words, the segment activity count code is not placed into CMR to initiate activity counting.

ECS AS A LINK MEDIUM BETWEEN MAINFRAMES

ECS can be used as a link medium device† between two mainframes (CX) as an alternative to 6683/6683 channel couplers (CC). To assemble the system with the ECS driver turned on, set IP.ECSLK nonzero. IP.ECSB must also be nonzero. The following ECSCOM installation parameters can be varied for the ECS driver.

IP.ECSLK	CEQU	0	0 = no MMF extended memory link, 1 = extended memory link can be used.
IP.LNKBF	CEQU	1	Number of MMF extended memory link buffers defined. One link buffer is required for each link between two distinct mainframes.
IP.MAXBL	CEQU	1461B	Size of maximum block to be transferred on one extended memory access. This cannot exceed 1461B.
IP.LNKMN	CEQU	500B	Minimum acceptable extended memory buffer length.
IP.ECYC	CEQU	37B	Controls primary restart cycle of the extended memory link driver. Driver restarts at primary cycle rate only if there is work to do. The following restart times are approximate in milliseconds.

3B	1 millisecond
7B	2 milliseconds
17B	4 milliseconds
37B	8 milliseconds
77B	16 milliseconds
177B	32 milliseconds
377B	128 milliseconds
1777B	256 milliseconds
3777B	512 milliseconds
7777B	1024 milliseconds

IP.CYSTP	CEQU	1	Controls the rate at which the link drive slows its restart rate when the link activity is low.
IP.EIDLE	CEQU	5	Controls the number of idle cycles allowed before the link decides to slow the restart rate due to inactivity.
IP.ECLNK	CEQU	0	0 = no simulation for dual test mode 1 = allows simulation of dual computers for extended memory link testing on one mainframe. IP.LNKBF must be set to 2.

†LCME and UEM cannot be used as link media between two mainframes.

UPDATE INSTALLATION OPTIONS

The following Update features are available or unavailable through assembly options and may be modified by deleting the appropriate entry in the range UPDATE.703 through UPDATE.711; specify these changes in the installation deck PL1CI.

DECLKEY	Enables DECLARE directive.
CHAR64	Supports full 64-character set.
PMODKEY	Enables PULLMOD card and G option.
AUDITKEY	Allows audit functions.
EDITKEY	Allows merge and edit.
EXTOVLP	Enables detection of four types of overlap involving two or more cards in a correction set.
DYNAMFL	Declares dynamic table expansion. When this option is assembled, Update automatically expands tables as required and dynamically requests the system 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.

An attempt to use features when the option has not been assembled causes Update to issue error messages. For example, when PMODKEY is not set, the PULLMOD card is not recognized as a legal directive.

All the above features are enabled by default.

COMMON MEMORY MANAGER VERSION 1 (CMM)

CMM provides control over all dynamic memory in the field length of a job. Its features are described in the Common Memory Manager Reference Manual. Products that use the CMM include the following.

- CYBER Loader 1
- COBOL 5
- Sort/Merge 5
- CDCS 2
- Query Update 3
- FORM 1
- FORTRAN Common Library 4

- FORTRAN Common Library 5
- BASIC 3
- COBOL Conversion Aids 4
- Data Base Utilities 1
- CYBER Record Manager (BAM and AAM)
- SYMPL 1
- FORTRAN Data Base Facility

CMM requires the same minimum hardware configuration as the operating system.

CMM uses symbol definitions from common deck CMMCOM. IPARAMS symbols, which specify the operating system, are also referenced. You can change the following CMMCOM installation parameters for CMM.

Name	Default	Description
DEFVER	0	Defines which CMM version is to be used by default. 0 A nonerror checking version (referred to as FAST) is used. 1 An error checking version (referred to as SAFE) is used.
FLF	2000B	If 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.

DEADSTART LOADING THE OPERATING SYSTEM

The operating system must be loaded before the computer can execute jobs. This procedure involves operator action depending on the type of magnetic tape unit available. The operator should be aware that when the deadstart button is pressed too long, multiple deadstarts might occur that can cause the deadstart tape to be read prematurely, possibly overwriting critical information. To avoid such a possibility, ensure that the deadstart tape is at the load point, and activate the deadstart button or switch briefly. Refer to the CYBER Initialization Package (CIP) User's Handbook for the correct procedure for warmstarting and coldstarting the system.

CONTROLWARE BINARY CREATION

COLDSTART DECK CREATION

Recreate the coldstart card decks by running the following job.

```
Job Statement.
REQUEST(OLDPL,E,HY) NOS/BE 1 PL1A
REWIND(OLDPL)
UPDATE(Q)
COMPASS(I=COMPILE,S=PPTEXT,L=0,B=PUNCHB)
7/8/9

*COMPILE ABC
6/7/8/9
```

This assembly will produce a small binary card deck. The coldstart card deck including the controlware is constructed as follows.

```
BINARY CARDS OF ABC
7/8/9
CONTROLWARE BINARY DECK (unprefixed or deck usable as input to COPBC)
6/7/8/9
```

WARMSTART DECK CREATION

New releases of controlware for magnetic tape and random mass storage controllers need a program name and a prefix (7700) table for use in the NOS/BE system. When controlware is on a CIP tape, use the GETCW procedure (also on the CIP tape) to move controlware to NOS/BE permanent files. Further details about the GETCW procedure are found in the CIP release materials. When controlware is on a controlware release tape, use COPBC. COPBC is a system program that uses an input directive record to add a prefix table and a program name to the controlware binary record. Select a program name and a permanent file name from the table below, and execute the example job to create controlware saved at the specified permanent file. DST1 and DST2 build jobs use these permanent files when creating NOS/BE deadstart tapes.

CONTROLWARE

Hardware	Identification Number	Program Name	Permanent File Name (pfn)
7021 (Tape)	MB434	OMT	OMTCWARE
7054 (RMS)	MA710	OSY	OSYCWARE
7154 (RMS)	MA401	OSZ	OSZCWARE
7155 (RMS)	MA721	OSJ	OSJCWARE

The following is an example job.

```
CWJOB,TO,PE1.
REQUEST(TAPE1,PE)                          controlware tape
REQUEST(BIN,PF)
COPBC.
CATALOG(BIN,pfn,ID=CWARE)
7/8/9
Select one group from the following directives for the prefix table and program name.
6/7/8/9
```

7021 Controller, program name = OMT

```
01  77000016000000000000
01  33152400000000000000
15  00000000000000000000
01  33152400000000000000
```

7054 Controller, program name = OSY

```
01  77000016000000000000
01  33233100000000000000
15  00000000000000000000
01  33233100000000000000
```

7154 Controller, program name = OSZ

```
01  77000016000000000000
01  33233200000000000000
15  00000000000000000000
01  33233200000000000000
```

7155 Controller, program name = OSJ

```
01  77000016000000000000
01  33231200000000000000
15  00000000000000000000
01  33231200000000000000
```

SAMPLE JOB FOR CREATION OF INSTALLATION CMRs AND CMR LIBRARIES

```
CMRS,I00,T0,MT1
COMMENT.  CONTROL STATEMENT SEQUENCE FOR 819 SUBSYSTEM
LABEL(PL1G,D=HY,L=PL1G,NORING,R)        (819 subsystem only)
UPDATE(Q,P=PL1G,C=C819                  (819 subsystem only)
UNLOAD(PL1G)                            (819 subsystem only)
LABEL(PL1A,D=HY,L=PL1A,NORING,R)
COMMENT. ASSEMBLE PPTEXT
UPDATE(Q,P=PL1A,C=PPTXT)
COMPASS(I=PPXT,L=0,S=CTITEXT,B=PPTEXT)
COMMENT.  START REPEATABLE CONTROL STATEMENT SEQUENCE, WHERE
COMMENT.  EACH REPETITION ASSEMBLES A NEW CMR.
UPDATE(Q,P=PL1A)
COMPASS(I,S=CTITEXT,B=CMRTEXT,L=0)                           CMRTEXT       (Extended memory
                                                                            system only)
COMPASS(I,G=PPTEXT,S=CTITEXT,L=0)                            CMR
COMPASS(I,G=PPTEXT,G=CMRTEXT,L=0,S=CTITEXT,B=CMRLIB)         CMR SEGMENTS  (Extended memory
                                                                            system only)
COMPASS(I=C819,S=PPTEXT,G=CMRTEXT,L=0,S=CTITEXT,B=CMRLIB)    819 SEGMENT   (819 subsystem
                                                                            only)
REWIND(CMRTEXT)
COPYBF(CMRTEXT,CMRLIB)
CATALOG(CMRLIB,CMRL108,ID=CCT,XR=XYZ,PW=XYZ,RP=20)
RETURN(CMRTEXT,CMRLIB)
COMMENT. END OF CONTROL STATEMENT SEQUENCE FOR CMRL108.
COMMENT. REPEAT ABOVE CONTROL STATEMENT SEQUENCE FOR ADDITIONAL CMRS
COMMENT. ADDING INPUT RECORDS AS APPROPRIATE.
COMMENT. 64 CMRS = MAXIMUM.
REWIND(LGO)
REQUEST(CMR,*PF)
REWIND(CMR)
COPYBF(LGO,CMR)
CATALOG(CMR,ID=CCT,XR=XYZ,PW=XYZ,RP=20)
7/8/9                                   (819 subsystem only)
*C LCM                                  (819 subsystem only)
7/8/9
*C PPTEXT
7/8/9
*/INPUT RECORD
*ID CMRL108
*I IPARAMS.15
*/IPARAMS MODIFICATIONS FOLLOW THIS STATEMENT
*/
*I CMRIP.1
*/CMR INSTALLATION PARAMETER MODIFICATIONS FOLLOW THIS STATEMENT
*/
*I CMR.1
*/ CMR MODIFICATIONS FOLLOW THIS STATEMENT
*/
*I LID.1
```

```
*/ LOGICAL ID TABLE MODIFICATIONS FOLLOW THIS STATEMENT
*/
*I EST.1
*/ EST CONFIGURATION FOLLOWS THIS STATEMENT
*/
*I RBR.1
*/RBR ENTRIES FOLLOW THIS STATEMENT
*/
*I FLAW.1
*/ FLAW ENTRIES FOLLOW THIS STATEMENT
*/
*/
*COMPILE        CMRTEXT,CWEOR1            (Extended memory system only)
*COMPILE        CMR
*COMPILE        CWEOR2.CWEOR3             (Extended memory system only)
7/8/9
6/7/8/9
```

The following constraints apply to building a multiple CMR deadstart tape. If a CMR has a separate extended memory library, the name of that library must be the same as that defined by IP.SLIB (CMRIP.1) in the corresponding CMR. Systems without extended memory do not need CMR segments and should disregard all statements applicable to extended memory and the 819 subsystem in the preceding example. Model 176 systems with the 819 subsystem hardware require all statements in the preceding example.

Examine the deadstart tape generation job DST1 (which needs extended memory defined) to realize how CMR handling is accommodated. DST1 may then be modified appropriately to capture the files created by job CMRS in the preceding example. For subsequent running system captures, jobs DST2 and DST3 can be modified to include additional CMR libraries.

SYSTEM RESIDENCY

The unconfigured deadstart tape contains 8DN, 8XS, 8XT, 8X8, 3DO, 4DO, A, 1SQ, AA, 1SP, and its 3SZ system device overlays as CM resident. These routines must be declared CM resident

Pages I-1-91 and II-1-92 have been removed.

when a deadstart tape is created. Additional routines such as CIO, 4ES, and 1AJ may be made CM resident to enhance system throughput. Routines established as CM resident in the running system will have CM residency on the new deadstart tape created by jobs DST2 and/or DST3.

Note that extended memory residency cannot be carried on a deadstart tape. Routines moved to extended memory resident will be established as disk resident on a new deadstart tape. The only way to set extended memory residency is by MOVE directive changes to the running system. None of the mentioned routines can be moved safely to extended memory.

System extended memory resident routines occupy a part of the paged area. The paged area is defined in terms of a page stack and accessed through a CMR central processor program (CP.CIO).

Extended memory residency cannot be specified via EDITLIB creation of the deadstart tape. An EDITLIB run must be performed after deadstart to move routines from their residence as loaded by deadstart to extended memory.

When EDITLIB creates a deadstart tape, it terminates the tape with a double end-of-file, effectively creating a null file following the last system library file on the tape. If this last file is not null during the preloading process (after system library files have been copied to mass storage), a job of the following structure is assumed.

```
Job Statement.
EDITLIB(SYSTEM)
7/8/9
EDITLIB Directives
6/7/8/9
```

This job will be copied to mass storage and cataloged as a permanent file with the following parameters.

```
LFN = ZZZZECS
PFN = ZZZZECS
ID  = SYSTEM (granted automatically for control point 0 permanent file operations)
TK  = SYSECSLIB
XR  = ECSLIB
```

This job will be run automatically by the terminate deadstart sequence PP program (TDS) whenever extended memory is in use and the deadstart level is either 0 (preload from tape) or 1 (load from the system permanent file).

Because of restrictions imposed for system (control point 0) permanent file operations, a user cannot catalog a new file with an ID of SYSTEM. Thus, ZZZZECS can be created only in the manner just described. Thereafter, the job can be modified by new-cycle catalog and old-cycle purges with the appropriate permissions (ID=SYSTEM allowed and required).

RHF SUPPORT; LOGFILE DEFINITION

The Remote Host Facility (RHF) and loosely coupled network (LCN) allow files in the input/output queue to be moved between mainframes. Permanent file LOGFILExxx (xxx is the physical ID) resides on the queue set and exists to handle routing information for files to be transferred between linked mainframes. The queue set does not require a PF device attribute on one member to allow cataloging LOGFILExxx.

A post-deadstart job executes program QLOG to attach and validate the LOGFILE. If it does not exist or problems are found when validating data, then a new LOGFILE will be created and cataloged.

ACCOUNTING FOR QLOG

If job accounting is needed to catalog a permanent file, then an ACCOUNT control statement must be added to the QLOG job. The following is an example:

```
*ID     QACCNT
*I      QLOGJOB.1
        CTLCD          (ACCOUNT Control Statement)
        CCEND
*C      GENDJ
```

QLOG job statements redefined in deck GENDJ on PL1A.

If job accounting is not needed to catalog a permanent file, no action is necessary.

STATION SUPPORT; SPOTS CORE REQUIREMENTS

The station control point MFSTAT handles all communication between the station and SCOPE 2; or between two CYBER 170, 6000, or CYBER 70 mainframes. In the following description of the station, the term 6000 refers to CYBER 170 Computer Systems, except model 176, and 6000 Computer Systems. The term 7000 refers to CYBER 70 Model 76 or 7000 Computer Systems. A spun-off task (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 spun-off tasks, permanent file SPOT, tape staging SPOT, spooling SPOT, dump SPOT, and deadstart SPOT.

The spooling SPOT is the only SPOT that transfers more than one file. The spooling SPOT is initiated when communication is established between mainframes and will not terminate until the station is dropped. The two SPOTs activated for spooling are SOT66 which transfers data between two 6000s, and SOT76 which transfers data between a 7000 and a 6000. All other SPOTs are initiated as required to transfer one file. Each SPOT is terminated after completion.

Field length requirements fluctuate as file transfers are initiated and terminated. As files are terminated, buffers are released. The lengths of the buffers used by the spun-off tasks are controlled by definitions in the common deck COMTUNE. Each SPOT uses one of the DEFs to determine the length of the I/O buffers. All spun-off tasks are written in the SYMPL language.

The nominal release definition values for the I/O buffers are the following.

```
LBUFDSPF = 1540=3004B (deadstart via permanent file)
LBUFDSTP = 1540=3004B (deadstart via tape)
LBUFDSLM = 4053=7725B (deadstart link buffer)
LBUFDP   = 2049=4001B (7000 dump)
LBUFLM   = 769=1401B  (link medium)
LBUFPF76 = 2601=5051B (permanent file for 6000-7000)
LBUFPF66 = 2081=4041B (permanent file for 6000-6000)
LBUFSP   = 1041=2021B (spooling)
LBUFTP   = 3079=6007B (tape staging)
```

Buffers required to transfer a file vary among the SPOTs. All SPOTs transfer a file between a 6000 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).

Table II-1-3. Core Requirements for Various SPOTS

SPOT Type	Code	Buffer Lengths Used to Transfer Files	Total Core Used†	Notes
Spooling				
6000-6000	3700B	LBUFSP+LBUFLM	13200B	Single file transfer
7000-6000	5100B	LBUFSP+LBUFLM	14600B	Single file transfer
Permanent File				
6000-6000	1600B	LBUFPF66	5700B	
7000-6000	2300B	LBUFPF76+LBUFLM	11000B	
Tape Staging				
7000-6000	3700B	3* MBL + LBUFLM/2+3	7600B	Maximum block length (MBL) as specified by user in tape staging. If MBL ≤ (LBUFTP-LBUFLM/2-3)/3 [7690]
		LBUFTP	11700B	If (LBUFTP-LBUFLM/2-3)/3 [7690] < MBL ≤ LBUFTP-LBUFLM-1 [23090]
		MBL + LBUFLM+1	20500B (MBL=50000)	If MBL ≥ LBUFTB-LBUFLM-1 [23090]
Dump				
7000-6000	1300B	LBUFDP	5300B	
Deadstart				
7000-6000	3300B	LBUFDSPF+LBUFDSLM	16300B	Via permanent file
		LBUFDSTP+LBUFDSLM	16300B	Via tape

NOTES:

Since spun-off tasks are like user jobs, they may be rolled or swapped out as the core is needed.

The relationship between buffer sizes and performance is bound by the same considerations as for any user job. The absolute minimum buffer size is a PRU + 2. Any large reduction in buffer sizes from the release values will have some impact on performance.

† As used with the nominal release definition values.

ACCOUNTING FOR THE SPOTS

Accounting for the SPOTs is handled through the common deck IPACCT on PL1D. To define accounting on the SPOT job statement, redefine the MICROs ACCTSP (for the spooling SPOT), ACCTTP (for the tape staging SPOT) and ACCTPF (for the permanent file staging SPOT), and leave the MICRO ACCOUNT as a null MICRO string. For accounting on an ACCOUNT statement, redefine the MICROs ACCTSP, ACCTTP, and ACCTPF as before and also redefine the MICRO ACCOUNT to the MICRO string ACCOUNT. For further information and examples, consult the common deck IPACCT. If accounting is not used (that is, IP.ACNT equals 0, no action is necessary.

The ACCTVAL micro in the common deck IPACCT provides a facility to allow account numbers to be validated and used for files that are cataloged by a SAVEPF from another mainframe. The ACCTVAL micro names an installation-defined program which can be called to validate the AC parameter from the SAVEPF statement that is passed to the SAVEPF SPOT job. The AC parameter is passed to the installation-defined program as a calling parameter. For example, if the ACCTVAL micro is defined to be VALIDAC, the following control statement will be placed in the SAVEPF SPOT job before the COMMENT.ON. statement.

```
VALIDAC,acparam.
```

acparam is the AC parameter value in the SAVEPF statement from the linked mainframe.

The installation program should validate the AC parameter and call a helper PP program to place it into W.CPFACT in the job control point area so that it will be associated with the file that is cataloged. If the AC parameter is invalid or absent, the program should issue a dayfile MESSAGE macro with the text ON. to turn on the dayfile transfer, then an appropriate diagnostic, then a dayfile MESSAGE macro with the text OFF. before aborting the SPOT job. The ON. and OFF. messages instruct the station to transfer the intervening messages to the connected mainframe.

The default value for ACCTVAL is COMMENT., so an account validation program will not be called unless ACCTVAL is redefined by the installation.

STATION

INSTALLATION PARAMETERS

A. TABLE SIZE - COMDECK COMTUNE-PL1D

NMF

The number of mainframes (6000 or 7000) to which the station may be linked.

DATAENT

The maximum number of active data streams.

SFTS

The size of the station file table is equivalent to DATAENT and used by the passive side only.

SNTS

The size of the SPOT name table (SNT). The size of the SNT should be the number of DATAENT plus the maximum number of local SPOTs (4) plus the spooling SPOT (1).

IDTMAX

The maximum size of the IDT table controls the number of logical IDs a mainframe may have and allows this value minus two logical IDs.

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 7000. The default value is set to 1, but it must be increased when the display improvements feature is selected.

SYNTAXSIZE

The size of the syntax extension table for the transparent display interface to the 7000. The default value is set to 1, but it must be increased when the display improvements feature is selected.

B. POLLING AND RECALL TIMES - COMDECK COMTUNE-PL1D

MSEC(I)

Used to set recall times. Since recall is given as 1/4 milliseconds, this function multiplies I by 4 so recall becomes I milliseconds.

SEC(I)

A way of approximating the number of busy station loops in I seconds.

ISEC(I)

Similar to SEC(I) but for the idle station.

TME7000

The delay in seconds of sending the time request to the 7000.

TME6000

The delay in seconds of sending the time request to the 6000.

STA7000

The delay in seconds used by MFSTAT between sending status requests to the 7000.

STA6000

The delay in seconds used by MFSTAT between sending status requests to the 6000.

LLV6000

The delay in seconds used by MFSTAT between sending load-leveling requests (applies to 6000 only).

RCL7000

The recall time in milliseconds used by the busy overlay of MFSTAT when communicating with a linked 7000 mainframe.

RCL6000

The recall time in milliseconds used by the busy overlay of MFSTAT when communicating with a linked 6000 mainframe.

TIMEOUT

The length of time in seconds used by MFSTAT 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.

LOCPASSTIME

The delay in seconds used by MFSTAT between looking for local GETPF/SAVEPF operations.

BSYLIM

The length of time the busy overlay of MFSTAT delays after sensing an idle condition before going into an idle state.

IDLRCLTM

The recall time used by MFSTAT when the idle overlay is executing.

NOTE

For better response time, lower both the RCL and STA values. To reduce CPU utilization, increase the RCL value. If the RCL and STA parameters are too greatly reduced, this may cause STD (the link medium coupler driver) to be locked in.

LOOPLIM

The delay in seconds used by MFSTAT between 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 MFSTAT waits for a reply before it rejects a DSD request.

MAXINCOUNT

The frequency with which MFSTAT calls QAC to check the input queues for files to spool when it is idle.

OVLMAX

The maximum time 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 delay in seconds used by MFSTAT after completion of spooling activity before going idle.

IDLETIME

The elapsed time in seconds after which the spooling SPOT attempts to initiate spooling operations.

IDLETIME2

The elapsed time in seconds after which the spooling SPOT initiates new spooling operations when output spooling is taking place.

SPOOLRCL

The recall time in milliseconds used by the spooling SPOT when there is no spooling activity.

C. BUFFER SIZE - COMDECK COMTUNE-PL1D

BUFSIZE

The size of the I/O buffer in the station control point on the passive side.

DAYBUFSIZE

The size of the MFSTAT buffer for processing spot dayfiles on the active side.

RRBUF

The size of the MFSTAT active transmit buffer, the passive receive buffer, the linked staged packet buffer, and the local stage packet buffer.

LRGBUF

The size of the MFSTAT receive buffer for the active side of the station.

LRGRBUF

The size of the MFSTAT transmit buffer for the passive side of the station.

LICRBUF

The length of the MFSTAT buffer used by the INTERCOM queue utility helper.

LBUFLM

The length of the link buffer used by SPOT jobs for 6000-7000 permanent file staging and 6000-6000 and 6000-7000 I/O spooling.

LBUFSP

The length of the disk buffer used by the spooling SPOT for 6000-6000 ad 6000-7000 spooling of I/O files.

LBUFPF66

The length of the buffer used to read and write the disk and link files for 6000-6000 permanent file transfers.

LBUFPF76

The length of the disk buffer used to read and write the disk for permanent file transfers to and from the 7000.

D. LOAD LEVELING - Input (6000 to 6000) DECK SSH-PL1A

Load leveling provides the capability of distributing the work load among linked 6000 mainframes with common logical IDs. Load leveling is used for jobs in the input queue with destination IDs (ST specified on the job statement) common to both linked mainframes.

A load-leveling algorithm (located in deck SSH near SSH.4350) determines whether or not load leveling is performed. The algorithm uses separate parameters (defined near SSH.311) for class 1 (allocatable) and class 2 (nonallocatable) jobs. Load leveling is performed independently for each class, depending upon these parameters.

Class 1 jobs require only immediately allocatable resources, such as memory and disk space. Class 2 jobs require additional resources that cannot be immediately allocated, because resources such as tape drives must be scheduled. For each class, the algorithm determines whether or not all of the required conditions for load leveling are satisfied by the mainframe on which the algorithm is called.

When user direct access (DA) ECS swapping is disabled (IP.ECSW=0), jobs that require DAECS are assigned to class 2. When IP.ECSW=0, jobs that require DAECS but no nonallocatable equipment are assigned to class 7. Maximum and current job counts and load-leveling parameters for class 1 also apply to class 7 jobs.

The following parameters are defined.

ICPFFNT

The number of free file name tables (FNTs) that must be available for a mainframe to accept a job.

ICPBJTA/ICPBJTN

The number of additional class 1/class 2 jobs that are allowed to execute must be greater than or equal to this value for a mainframe to accept a class 1/class 2 job.

ICPRJTA/ICPRJTN

The number of class 1/class 2 jobs in the input queue that are ready to run must be less than or equal to this value for a mainframe to accept a class 1/class 2 job.

The job control area contains the mainframe status information used by the algorithm to determine whether or not load leveling is to be performed. (Byte C.JCA Of pointer P.SCH points to the job control area.)

For class 1 jobs, the load-leveling conditions are as follows.

- The number of available FNTs > ICPFFNT
  (The number of available FNTs is obtained from C.JCEMC.)

- The number of additional class 1 jobs allowed to execute ≥ ICPBJTA
  (The number of additional class 1 jobs allowed to execute is C.JCMXB-C.JCCNB; that is, the maximum number of class 1 jobs allowed to execute minus the number currently in execution equals the number of additional class 1 jobs allowed to execute.)

- The number of class 1 jobs in the input queue that are ready to run ≤ ICPRJTA
  (The number of class 1 jobs in the input queue that are ready to run is obtained from C.JCNJI.)

For class 2 jobs, the load-leveling conditions are as follows.

- The number of available FNTs > ICPFFNT
  (The number of available FNTs is obtained from C.JCEMC.)

- The number of additional class 2 jobs allowed to executive ≥ ICPBJTN
  (The number of additional class 2 jobs allowed to execute is C.JCMTB-C.JCCTB; that is, the maximum number of class 2 jobs allowed to execute minus the number currently in execution equals the number of additional class 2 jobs allowed to execute.)

- The number of class 2 jobs in the input queue that are ready to run ≤ ICPRJTN
  (The number of class 2 jobs in the input queue that are ready to run is obtained from C.JCNTJ.)

When the system invokes load leveling, the sending mainframe transfers jobs with destination IDs to the receiving mainframe for processing. The following conditions must exist on both mainframes for load leveling to be performed.

- Any of the three conditions (for the class being considered) on the sending mainframe is false. (This is normally the case, since the number of free FNTs is usually greater than ICPFFNT.)

- All three conditions on the receiving mainframe are true. (This occurs if enough FNTs are free, enough additional jobs are allowed to go into execution, and only a few jobs are in the input queue; that is, the mainframe can accept more jobs.)

E. LOAD LEVELING - Output (6000 to 6000) COMDECK COMTUNE - PL1D

Load leveling for these types of output files takes place only for files with DIDs shared by both machines if there is more than this number for a particular type in the output queue.

ICPTPRT (5)

The load-leveling threshold value for print files.

ICPTRUN (5)

The load-leveling threshold value for punch files.

ICPTOTH (5)

The load-leveling threshold value for types of files other than print and punch files, such as microfilm, plotter, and so forth.

F. DSD PARAMETERS - PL1A

IP.LINK

The number of links that can be connected to the station. Used in text IPTEXT, defined at IPARAMS.15.

IP.7LNK

For symmetric (6000 to 7000) or reverse (7000 to 6000) staging, IP.7LNK (within DSD at DSD.220) must be defined as not equal to zero; and the symmetric-reverse staging group (SYSTG) (SCOPE 2.1) must be LCM-resident. If SYSTG is disk-resident, then IP.7LNK must equal zero.

IP.STEX

For use of the display improvements interface (which uses the message codes MC.LSYN, MC.LCOM, and MC.LDIS), this parameter must be specified in DSD at IPARAMS.15 as unequal to zero.

G. ADDITIONAL PARAMETERS

Refer to COMTUNE and COMTUNX (two common decks on PL1D) for a detailed description of additional station tuning and configuration parameters. The minimum, maximum, and default values are defined together with notes and cautions on each variable.

For use of extended memory as a link medium device instead of the 6683-6683 remote couplers on a 6000 to 6000 link, refer to ECS as a Link Medium Between Mainframes in this section.

ACCOUNTING FOR GENLDPF-GENERATED JOBS

GENLDPF generates jobs to selectively load permanent files that had a permanent file catalog entry at the time PFLOG was run. Accounting for the generated jobs is handled through the common deck COMSLOG on PL1B. Generated job and account statements are located at Update identifiers GENACNT.1 and GENACNT.2, respectively. Refer to common deck COMSLOG for complete directions for modification of these statements. If accounting is not used (that is, IP.ACNT = 0), no action is necessary.

GEMINI LOAD LEVELING

Gemini, a control point program, provides input and output file load leveling between two linked mainframes under the control of NOS/BE. An RMS queue set must be shared between the two mainframes. This set or another shared set must be configured to hold permanent files.

Gemini at a control point on one mainframe communicates with Gemini at a control point on the other mainframe through two files cataloged on a shared RMS set (not necessarily the shared queue set). These files contain sections for communication of the following information.

- Contents of the IDT (mainframe and logical identifier table).
- Message from the sending mainframe (used for file transfer).
- Message acknowledgment from the receiving mainframe.
- Contents of the input, output, punch, special, and execution queues in QAF format (refer to the NOS/BE System Programmer's Reference Manual).
- Common LID input and output queue counts (common LIDs are those that appear in the IDT for both mainframes).

Only the ownership of the input or output queue files is transferred between the two mainframes because the files exist on a shared queue set. The messages are pointers to the appropriate catalog entry being transferred. The IDT information distinguishes between LIDs unique to one mainframe and those common to the two mainframes. The IDT information is updated whenever Gemini detects a change in the IDT. Queue counts of common LIDs are kept to allow Gemini to determine when load leveling is required. The contents of queues for a mainframe allow INTERCOM users to locate their files through the MYQ/Q/FIND utilities.

INSTALLATION PARAMETERS

You can change the following parameters by making Update insertions at GEMIPRM.7. Default values are shown in parentheses following the parameters.

```
N$MF        CEQU        n            (1)
```

Number of mainframes in Gemini network; n can be 0, 1, or 2. A value of 0 or 1 informs the routine MYQ that the Gemini link is not available. N$MF must be equated to 2 if Gemini is to be used.

```
PID$MFA     CMICRO      3,/xxx/      (MFA)

PID$MFB     CMICRO      3,/yyy/      (MFB)
```

xxx and yyy are physical identifiers (PIDs) of the two mainframes in the network. They must be three letters and must match the PID in the IDT.

```
PID$PFNA    CMICRO      n,/xx.../    (LINKFILE)

PID$PFNB    CMICRO      n,/yy.../    (LINKFILE)
```

Permanent file names of the two link files in the network (one through forty characters; n is the length of string xx... or yy...).

```
PID$IDA     CMICRO      n,/xxx.../   (MFASYSTEM)

PID$IDB     CMICRO      n,/yyy.../   (MFBSYSTEM)
```

Permanent file identifers (one through nine characters; n is the length of string xxx... or yyy...) of the two link files in the network. If these parameters are not specified but the PID$MFx micros are specified, the defaults are the characters SYSTEM prefixed with the PID$MFx parameters.

```
LINK$AC     CMICRO      n,/xxx.../   (123456789)
```

Account number (one through nine characters; n is the length of string xxx...) of the two link files in the network.

```
LINK$SN     CMICRO      n,/xxx.../   (*PF)
```

Set on which the link files reside. Possible values for xxx... are *PF, *Q, *SYS, or the actual setname (one through seven characters; n is the length of string xxx...). The link files must reside on a shared RMS set.

```
LL.RCL      CEQU        n   (6)
```

Gemini recall count; delays Gemini execution. When Gemini finds unique LIDs with files to transfer, it loops every half second to complete a portion of the transfer process. When nothing remains to be done and there are no more files to transfer, Gemini waits LL.RCL * .5 seconds before looping, thus saving system resources. The queue information for one queue on the link file is updated every LL.RCL * .5 seconds; therefore, it takes 5 * LL.RCL * .5 seconds to update all queue information on the link file. Gemini attempts to initiate load leveling for common LIDs every 5 * LL.RCL * .5 seconds. LL.RCL can be used to adjust the balance between Gemini processing and system utilization. To increase Gemini processing, reduce the value; to reduce system utilization, increase the value.

LL.MINI	CEQU	n	(2)
LL.MINO	CEQU	n	(2)

Minimum count for load-leveling input (I) and output (O) files (refer to Load-Leveling Algorithm).

LL.MAXI	CEQU	n	(6)
LL.MAXO	CEQU	n	(6)

Maximum count for load-leveling input (I) and output (O) files (refer to Load-Leveling Algorithm). LL.MAXx should be at least three times LL.MINx to avoid transferring too many files.

LOAD-LEVELING ALGORITHM

Gemini makes load-leveling decisions based upon comparisons of input file counts for one common LID at a time. (Output file count includes all files with a disposition code of 10 or greater.)

Gemini transfers LL.MINx files for a common LID to the second mainframe if the first mainframe has at least LL.MAXx files and the second mainframe has fewer than LL.MINx files. Gemini also transfers one file for a common LID to the second mainframe if the first mainframe has at least two files but fewer than LL.MAXx files and the second mainframe has no files.

In addition, Gemini transfers files with a LID that exists on only one mainframe. It scans tables for this type of transfer every LL.RCL * .5 seconds whenever it is not busy load-leveling files.

NOTES AND CAUTIONS

Gemini does not discriminate among job classes when the input files have the same LID. Gemini does not check for dependency IDs when load leveling is executing. It is suggested that you set up LIDs based on job statement parameters such as the following:

Time use
Memory use
Extended memory use
Tape use
RMS use
I/O use
Priority level
Dependency

The use of dependency requires a unique LID. The use of tapes might also require a unique LID. The other parameters could be grouped and given a unique LID or a common LID, depending upon your goals.

Gemini and the station can be run on the same system at the same time. However, the station predominates over Gemini in transferring jobs.

DEADSTART DUMP ANALYZER

PL1B and PL14 contain the fast dump analyzer. The overlay on PL14 is for INTERCOM analysis only. If INTERCOM 5 is not built, an analyzer without the capability to format INTERCOM tables can be produced by inserting the following directive after the line that states ADD MODIFICATIONS HERE in the second input record of PL1BI:

```
*DEFINE NOINT
```

This modification removes IT as a default parameter value and the resulting analyzer flags the IT analyzer option as an error. All other options are unaffected.

The directions for taking an express deadstart dump (EDD) appear in the CYBER Initialization Package (CIP) User's Handbook, and the directions for using the analyzer appear in the NOS/BE System Programmer's Reference Manual.

ACCOUNTING FOR CONTROLWARE DUMP FILES

Auto recovery of 7155 controlware saves a copy of the hung controlware on the system set using permanent file name:

```
CONTROLWARE7155hidCHnn,ID=DMP7155
```

hid Host mainframe identifier.

nn Channel number.

If specific accounting is needed for the cataloged controlware dump, then the accounting parameter needs to be added to the FCP routine in deck 1SQ on PL1A. If no specific accounting is required, no action is necessary.

COMPASS VERSION 3.6 AND COMMON COMMON DECKS 2

RELEASE DESCRIPTION

COMPASS is a comprehensive assembler program for the CYBER 180, CYBER 170, CYBER 70, 6000, and 7000 Computer Systems. COMPASS 3 runs under NOS/BE 1 and requires the same minimum hardware configuration as NOS/BE 1.

The common common decks are a set of debugged COMPASS subroutines that perform such functions as:

- Data conversion.
- Dynamic table management.
- Register saving and restoration.
- I/O interface with CIO and FET.

RELEASE MATERIALS

The release tape for COMPASS, PL2, contains a source program library as file 1 and assembled binary as file 2.

GENERAL DESCRIPTION

COMPASS consists of three overlays. The level (0,0) overlay COMPASS is the main control program. The level (1,0) overlay COMP3$ contains the assembler which can be called by compilers to process embedded COMPASS source programs. The level (1,1) overlay COMP3$A contains the part of the assembler that is loaded after initialization is complete.

The common common decks are available as UPDATE COMDECKS in source form on the COMPASS oldpl or as relocatable subroutines in the SYSLIB library.

INSTALLATION PARAMETER

To ensure efficient code generation, set the MODEL micro in deck IPARAMS on PL1A to the proper value for the target machine.

COMPASS has one installation parameter, CP.OVLIB, the library name for overlays. In the released system, overlays COMP3$ and COMP3$A must be in a library in the global library set for a job or in the NUCLEUS library. COMPASS loads its overlays from a specific library with the following.

```
*D CPS028.10
CP.OVLIB     MICRO 1,,*libname*
```

Changing this parameter from its default state necessitates change in the installation deck EDITLIB record. Note that this change is needed only when the overlays are to be in a system library other than NUCLEUS. If the micro CP.OVLIB is left null (as released), COMPASS can be executed without change or reassembly from the system library NUCLEUS, from any system or user library named on a LIBRARY control statement, or from an overlay (nonlibrary) file.

INSTALLATION PROCEDURES

Installation of COMPASS and the common common decks requires obtaining job decks PL2I and PL2E from the installation deck program library tape as outlined in part I, section 1 of this document.

PL2I is a maintenance deck used to create a revised program library and binary file. PL2E can be used to enter COMPASS and the common common decks into the running system or user libraries from either the released PL2 or a tape created by PL2I. After deck PL2E has completed, run job DST3 to create a deadstart tape of the running system. You need not run job decks PL2E and DST3 if the user library installation process is being followed.

The installation of COMPASS includes cataloging a permanent file called COMCPL, ID=CCT, which is a program library containing only the common common decks. The site analyst is responsible for making this file accessible to users of COMPASS absolute assemblies.

CYBER RECORD MANAGER 1 BASIC ACCESS METHODS 3

RELEASE DESCRIPTION

CYBER Record Manager Basic Access Methods (BAM) consists of modules for creating, updating, and accessing two file organizations: sequential (SQ) and word-addressable (WA).

BAM 1.5 operates under NOS/BE on the same minimum configuration as NOS/BE.

The structure of the release tape PL3A is as follows.

Files	Content
1	Program library in UPDATE format
2	TXTCRM, IOTEXT, SYSTEXT binary
3	Control modules binary
4	Encapsulated modules binary
5	FILE, CRMEP control statement processor relocatable binary
6	FILE, CRMEP control statement processor absolute binary
7	FORTRAN Extended 4 and FORTRAN 5 interface binary

NOTES AND CAUTIONS

The display option on parity errors is not implemented.

If C-blocked, non-W record, SI tapes are copied to S tapes, section boundaries may be lost.

End of block padding is not supplied on last block (short block) of each partition.

ADDITIONAL INFORMATION

SQ/WA I/O modules are divided into two parts: Initialization Modules and Sequential and Word Addressable I/O Modules.

INITIALIZATION MODULES

These routines control selective loading based on file organization. They contain jump vectors directing a user call to the I/O appropriate to the file organization selected. Their program names have an RM suffix.

SEQUENTIAL AND WORD ADDRESSABLE I/O MODULES

The I/O macro text included with the SQ/WA program library is IOTEXT, which is identical to the default SYSTEXT. It consists of, but is not limited to the macros included in the following table. (Some auxiliary macros exist which are not supported at the user level.)

Macro Name	System	Reference	Comdeck
FILE	CRM	CYBER Record Manager Reference Manual	CRMCOM
FETCH			
FLUSHM			
STORE			
OPENM			
CLOSEM			
GET			
GETP			
GETN			
GETNR			
GETWR			
PUT			
PUTP			
PUTWR			
REPLACE			
DELETE			
ENDFILE			
SKIPdu d=F/B,u=L/P/F			
SEEK			
START			
REWINDM			
WEOR			
WTMK			
GETL			
PUTL			
CLOSEL			
CHECK			
ABORT	NOS/BE 1	NOS/BE 1 Reference Manual†	ACTCOM
CHECKPT			
CLOCK			
CONTRLC			
DATE			
DISPOSE			
ENDRUN			
FILESTAT			
IOTIME			
IXi Xj/Xk			
IXi Xj/Xk,Bn			
JDATE			
LOADREQ			
MEMORY			
MESSAGE			
RECALL			
RECOVR			
REQUEST			
ROUTE			
RTIME			
STATUS			
SYSCOM			
SYSTEM			
TIME			
TRANSR			

† These macros are source compatible with the corresponding macros on CPCTEXT but they do not generate the same code.

INSTALLATION PARAMETERS

To ensure correct code generation, the MODEL micro in deck IPARAMS on PL1A must be set to the correct value for the target machine.

The following installation parameters permit a certain amount of tailoring.

Parameter	Mnemonic Update ID	Description	Default Value
*DELETE LBLIM.1	LBLIM	Length of label buffer and size limit of user label string. Each user label requires 9 words. LBLIM should be n*9+1, where n is the maximum number of labels permitted (HDR1-9,...).	10

Use of the compare/move unit (CMU) instructions in the routine MOVE.RM is affected by the definition of CMU and NOCMU in TXTCRM. For records over 40 characters, the CMU hardware reduces CP time of a program using SQ/WA.

To remove the CMU code:

```
*D,SWIA536.26
*C,TXTCRM
```

To remove the NOCMU code:

```
*D,SWIA536.27
*C,TXTCRM
```

If CMU and NOCMU are both defined (default), CRM will decide at execution time which move routine to use by checking the CMU flag in word RA.CMU.

INSTALLATION PROCEDURES

File 1 of PL3A contains the SQ/WA program library.

Files 2 through 7 are preassembled binaries assembled with default installation parameters.

Installation of SQ/WA requires obtaining job deck PL3A1, PL3AE, and PL3AO for the installation deck program library tape, as outlined in part I, section 1.

Deck PL3A1 references IPTEXT. Part III of this document contains a cross-reference map of IPARAMS symbols and routines that reference these symbols.

Deck PL3A1 is a maintenance deck which allows regeneration of PL3A. This deck updates the program library, assembles SQ/WA, and places the binary on the new PL as supplemental files. Modify user-selected installation parameters at the indicated place in PL3AI. Deck PL3AI requires access to the NOS/BE 1 program library PL1A to acquire the common decks ACTCOM and COMSRAS used by the CRM system texts.

Deck PL3AE adds to SQ/WA to the running system or user libraries, either from the released PL3A or a PL3A created by deck PL3AI. Then run deck DST3, described in section 1, to create a deadstart tape of the running system.

CYBER RECORD MANAGER 2 ADVANCED ACCESS METHODS 4

RELEASE DESCRIPTION

CYBER Record Manager Advanced Access Methods (AAM) includes modules for creating, updating, and accessing Indexed Sequential (IS), Direct Access (DA), Actual Key (AK), and Multiple Index Processor (MIP) files.

Four utility routines called by control statements are provided for AAM files. FLSTAT prints the statistics for existing files; FLBLOK produces estimates of block and buffer sizes from input statements; MIPGEN inverts an existing data file on any number of alternate keys and produces or modifies the associated MIP file; MIPDIS disassociates or reassociates MIP and data files.

A key analysis utility routine (KYAN) is available to aid in the selection of a hashing routine for direct access files.

A create utility (CREATE) is available for efficiently creating DA files.

Both KYAN and CREATE utilities require CMM to be available.

RELEASE MATERIALS

IS and MIP are contained on release tape PL3C.

The structure of the release format PL3C tape is as follows:

File	Content
1	Program library in UPDATE format
2	Encapsulated I/O modules
3	Absolute binaries for the AAM Extended utilities
4	Binaries for the AAM Extended relocatable programs
5	Relocatable binaries for MIPGEN
6	Relocatable binaries for FLSTAT
7	Relocatable binaries for FLBLOK
8	Relocatable binaries for MIPDIS

LIMITATIONS

The CREATE, IXGEN, and MIPGEN utilities require that Sort/Merge 5 be installed. If Sort/Merge 5 is not available, create comparable DA and multiple-indexed files through explicit CRM calls.

AAM INSTALLATION PARAMETERS

When installing AAM, code to gather additional file statistics is assembled if the UPDATE directive *DEFINE STATS is included in the UPDATE input to the AAM program library. If this directive is omitted, only normal file statistics are gathered.

USER ADDITIONS TO AAM

AAM includes one system compression/decompression routine. You can add up to 53 additional user written compression/decompression or encoding routines as system routines. Each added routine must be encapsulated and the capsule OPEN$AA must be modified. The procedure to add routines follows.

The routine must have one entry point whose name is of the form CMPR$nn, where nn is two decimal digits in the range 11 through 63. The entry point name of the first routine added must be CMPR$11, the second must be CMPR$12, and so on. The entry point must be the second word (word 1) of the routine.

The first three words of each routine must have the following format.

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

The following illustrates the construction of a single site-added compression/decompression routine.

```
        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, the VFD statements must be constructed as shown in the preceding example. Execution of the compression or decompression code is effected by a return jump to the address specified in word 1 or word 2 of the routine.

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

GENTBL epname

epname Entry point name specified in word 0 of added routine

Specify table entries in consecutive, ascending numerical order. For example, if three routines are added, the following change to OPNMDAA must be made.

```
*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 sequence of control statements including the following.

```
.
.
.
UPDATE (K)
.
.
.
COMPASS(I,S=TXTCRM,S=IPTEXT)            Assembles OPNMDAA and DICODAA
SYMPL(I,LXR)                            Compiles OPENDAA
COMPASS.                                Assembles user-written routine
.
.
.
GROUP($AAM$$CTL$)                       Encapsulates the modified OPNM$AA capsule
CAPSULE($OPNM$$AA$)                     and the new compression capsule
CAPSULE($CMPR$$11$)
LDSET(OMIT=$SETUP,$/$RM$$SYS=$)
LOAD,LGO.
NOGO,NEWCAP.
EDITLIB,SYSTEM.
7/8/9
*IDENT
.
.    UPDATE directives to modify OPNMDAA
.
*C OPNMDAA,DICODAA,CWEOR1,OPENDAA
7/8/9
.
.    User routine being added
.
7/8/9
READY(SYSTEM)
LIBRARY(AAMLIB,OLD)
REPLACE(*,NEWCAP)
FINISH.
COMPLETE.
ENDRUN.
7/8/9
6/7/8/9
```

INSTALLATION PROCEDURES FOR AAM

PL3C contains eight files. File 1 contains the program library which will generate COMPILE file records.

Source Language	Description
SYMPL	Code for Extended IS, DA, AK, and Extended MIP CP routines
COMPASS	Code for Extended IS, DA, AK, and Extended MIP CP routines
COMPASS	Code for MIPGEN utility
COMPASS	Code for skeleton file used by COPYL to rearrange binaries for capsule formation
FORTRAN Extended 4	Code for FLBLOK utility
SYMPL	Code for FLSTAT utility
COMPASS	Code for FLSTAT utility
SYMPL	Code for MIPDIS utility
COMPASS	Code for MIPDIS utility
COMPASS	Code for static load subroutines

PL3CI is a maintenance deck which can be used to create a revised program library and binary file containing modifications. PL3CE can be used to enter AAM Extended into the running system or user libraries, either from the released tape or a tape created by PL3CI. Job DST3 can be run to capture a deadstart tape containing AAM Extended. Decks PL3CE and DST3 need not be run if the user library installation process is being used. In non-ULIB mode, PL3CI executes an EDITLIB to the running system AAMLIB library to make required routines available during further overlay generation. Deck PL3CO allows regeneration and replacement of the absolute overlay file on tape PL3C, using the tape written by PL3CI.

NOTE

If FORM calls the KYAN utility from an owncode exit, KYAN must be available in the system.

VERIFICATION PROGRAMS

Install FORTRAN Extended 4 before running the corresponding installation verification. Comment statements describe the purpose of each deck. Two verification programs are provided.

8-BIT SUBROUTINES VERSION 1.1 5

RELEASE DESCRIPTION

8-Bit Subroutines run under NOS/BE and the CYBER Record Manager.

RELEASE MATERIALS

The 8-Bit Subroutines are released on release tape PL4 together with FORM. A complete catalog of PL4 contents follows.

Files 1 through 3	8-Bit Subroutines
1	Program library in Update format
2	8-Bit Subroutines binary capsules
3	COPY8P absolute binary
Files 4 through 7	FORM
4	Program library in Update format
5	Relocatable capsules (run time system)
6	FORM main overlay relocatable binary
7	FORM main overlay absolute binary

HARDWARE CONFIGURATION

The 8-Bit Subroutines require the same minimum hardware configuration as NOS/BE. An extended print train is required to print ASCII 96-character graphic files, if used.

GENERAL DESCRIPTION

The relocatable routines from the 8-Bit Subroutines run under NOS/BE and CYBER Record Manager with COBOL, FORTRAN, or COMPASS. COPY8P, a stand-alone routine used to print 360/370 files, can be called from a COPY8P control statement and runs under NOS/BE.

INSTALLATION PROCEDURES

Part III contains a cross mapping of referencing routines and IPARAMS symbols found in IPTEXT.

Obtain installation job decks PL4AI, PL4AE, PL4AV1, and PL4AV2 from the Installation Deck program library using the procedure described in part I, section 1.

Deck PL4AI is a maintenance deck that allows updates of the 8-Bit Subroutines on the PL4 tape. This deck updates the program library, assembles the relocatable object routines, assembles COPY8P, and creates a new COPY8P absolute overlay. The job allows creation of a revised PL4 release tape.

Deck PL4AE adds the 8-Bit Subroutines to the running system. Relocatable object routines are put in the BIT8LIB library. COPY8P becomes part of the NUCLEUS library. Deck DST3 then can be run to create a deadstart tape of the running system. Decks PL4AE and DST3 need not be run if the user library installation process is being followed.

FORM VERSION 1.1 6

GENERAL DESCRIPTION

FORM copies and restructures data files. It can be called from a FORM control statement.

RELEASE MATERIALS

FORM is contained on release tape PL4 with the 8-Bit Subroutines. A complete catalog of the PL4 contents appears in part II, section 5.

HARDWARE CONFIGURATION

FORM requires the same minimum hardware configuration as NOS/BE.

INSTALLATION PARAMETERS

No IPARAMS are used.

PL4 contains seven files; files 4 through 7 pertain to FORM.

Obtain installation job decks PL4BI, PL4BE, and PL4BO from the Installation Deck program library by using the procedure described in part I, section 1.

Deck PL4BI is a maintenance deck that allows updates of FORM on the PL4 tape. This deck updates the library, assembles and encapsulates the relocatable routines, and assembles and builds a new FORM main overlay. The job allows creation of a revised PL4 release tape.

PL4BE adds FORM to the running system. Relocatable capsules are placed in the BIT8LIB library. FORM is placed in the NUCLEUS library.

Deck PL4BO allows regeneration and replacement of the absolute overlay file on tape PL4.

ON-LINE MAINTENANCE SOFTWARE 7

RELEASE DESCRIPTION

The On-Line Maintenance Software (OLMS), previously known as the CE Diagnostics, requires the same hardware configuration as NOS/BE.

RELEASE MATERIALS

The On-Line Maintenance Software is released on the release tape PL5A.

The structure of the tape containing OLMs programs is as follows:

File	Content
1	Program library in UPDATE format
2	PP COMPASS binaries
3	CP COMPASS binaries
4	FORTRAN binaries

INSTALLING DIAGNOSTIC PROGRAMS

Installation of this product requires the Update directive *DEFINE, names to write correct data to the COMPILE file and the F parameter on the COMPASS statement to control code assembly for proper binary creation. To include test CMS for a mainframe with a compare/move unit, place a *DEFINE,CMU directive in job PL5I.

DEFINE Directive	Required Driver
*DEFINE,SECURE	Read preallocated area only unless disk is logically idle or unloaded

The COMPASS F parameter must be equated to 4 (F=4) when assembling CP COMPASS programs. The FORTRAN 5 compiler is required for compiling FORTRAN programs.

INSTALLATION PARAMETERS

Release values of the installation options in the deadstart diagnostic sequencer are as follows:

NOISEL (DDS.194)

Same as IP.NOISE in deck CIOCOM on PL1A; default setting is 3.

INSTALLATION PROCEDURES

Installation of the On-Line Maintenance Software requires deck PL5AI from the installation deck program library as outlined in part I, section 1.

Deck PL5AI references IPTEXT; part III of this document contains a cross-reference map of IPARAMS symbols and routines that reference these symbols.

Installation job PL5AI updates and creates a new program library including assembled/compiled binaries.

CDC licensed maintenance customers should refer to the Concurrent Maintenance Library (CML) Version 3.2 Reference Manual for CML installation procedures. CML installation applies to CDC licensed maintenance customers only.

MAINTENANCE PACKAGE 8

RELEASE DESCRIPTION

Maintenance tools for NOS/BE are provided on release tape PL6. These maintenance tools are divided into two categories: SYMPL compiler and conversion aids. The structure of the release format PL6 is as follows.

Files 1 through 4	SYMPL
1	SYMPL source in Update program library format
2	SYMPL compiler in relocatable binary
3	SYMPL compiler in absolute overlay binary
4	SYMPL object library in relocatable binary

Files 5 through 6	Conversion aids
5	Conversion aids source in Update program library format
6	Conversion aids binary

SYMPL

SYMPL (Systems Programming Language) is designed to facilitate systems programming; it does not contain some features normally found in higher level languages, such as complex arithmetic and input/output capability. Instead, it contains features particularly suited to systems programming, such as bit manipulations, based arrays, and an elementary macro capability. It produces code optimized for efficient register and functional unit usage, particularly oriented toward the 6600-type mainframe.

The SYMPL Compiler is written mainly in SYMPL; only the system interface routines are in COMPASS. Thus, an alsolute binary of SYMPL is necessary for installation if changes are to be made to the source.

INSTALLATION PARAMETERS

SYMPL has no installation parameters. However, to ensure efficient code generation, set the MODEL micro in deck IPARAMS on PL1A to the proper value for the target machine.

INSTALLATION PROCEDURES

Before SYMPL can be installed, NOS/BE, COMPASS, and the FORTRAN Extended 4 object library must be installed. Update and install SYMPL with the following jobs. Job PL6AI updates the SYMPL library tape. Job PL6AE edits SYMPL into the system from a SYMPL program library tape. Deck PL6AO allows regeneration of absolute overlays plus creation of a new PL6 tape. Because SYMPL is written in its own language (SYMPL), it is recommended that the latest available SYMPL binaries be present in the running system before installing SYMPL. The unconfigured deadstart tape base system will always contain the latest SYMPL binaries; all other base systems should enter the latest available SYMPL binaries into the running system using deck PL6AE.

VERIFICATION PROGRAM

The best verification of successful installation of SYMPL is satisfactory compilation of FORM and Query Update.

CONVERSION AIDS

The following conversion aid programs are provided on the maintenance tools tape.

SIFT	A program to convert FORTRAN 2.3 programs to FORTRAN Extended 4 format
SPY	Utility package used to monitor the P register of a CPU program and provide a histogram of elapsed time used in specific areas of code
DOCK	Utility for extracting IMS information from the NOS/BE program library
SIS63	A program to convert a SIS file created on a SCOPE 3.3, 63-character set system, so that it can be processed on a NOS/BE, 64-character set system
CIA	A PPU program used to collect data on CPU utilization or CPMTR execution for performance analysis
CPMET	A program to collect and report detailed data on CPMTR execution

INSTALLATION PROCEDURES

Job PL6BI updates and compiles file 5 creating a complete revised PL6. Job PL6BE adds binary from file 6 to the running system. Obtain these jobs from the installation deck program library tape, using the procedure outlined in part I, section 1.

Once PL6BE has completed, run job DST3 to create a deadstart tape of the running system. PL6BE and DST3 are not required if the user library installation process is being used.

USE OF CONVERSION AIDS

Use instructions for SIFT can be found in SIFT (FORTRAN Translator Program) PSB.

Use instructions for the other conversion aids are as follows.

SPY USE

SPY is a peripheral processor program that collects sample P register values during the execution of a central processor (CPU) program. While the CPU program is running, SPY reads the P register at intervals of 34 microseconds† and groups its samples according to three basic parameters:

low	Octal address of the beginning of the range within which addresses are counted.
high	Octal address of the end of the range within which addresses are counted.
binw	The octal number of program addresses that are grouped together for data collection. P register values are recorded in sequential ranges of this size; must be a power of 2 and $1 \leq$ binw $\leq$ 100B.

Each grouped collection of P register values is called a bin. The number of bins (nbins) that is processed is (high-low)/binw. This number must not exceed the number that SPY is capable of processing (approximately 2040B). If this limit is exceeded, SPY automatically reduces the effective value of parameter high to bring the number of bins within the range of its capability.

On each reading of the P register, if the register's value is within the range low $\leq$ P < high, 1 is added to the bin number (P-low)/binw.

Two special bins are maintained to count the P register values that are less than parameter low or that are greater than or equal to parameter high.

Two other special bins are maintained to count time while the CPU program is not in execution. One of these counts time periods while the program is in recall status, and the other counts periods while the program is in waiting status. Waiting status is when the program is not in recall, but is not executing because the CPU has been assigned to a different job; this includes the time while monitor mode programs are executing.

Output

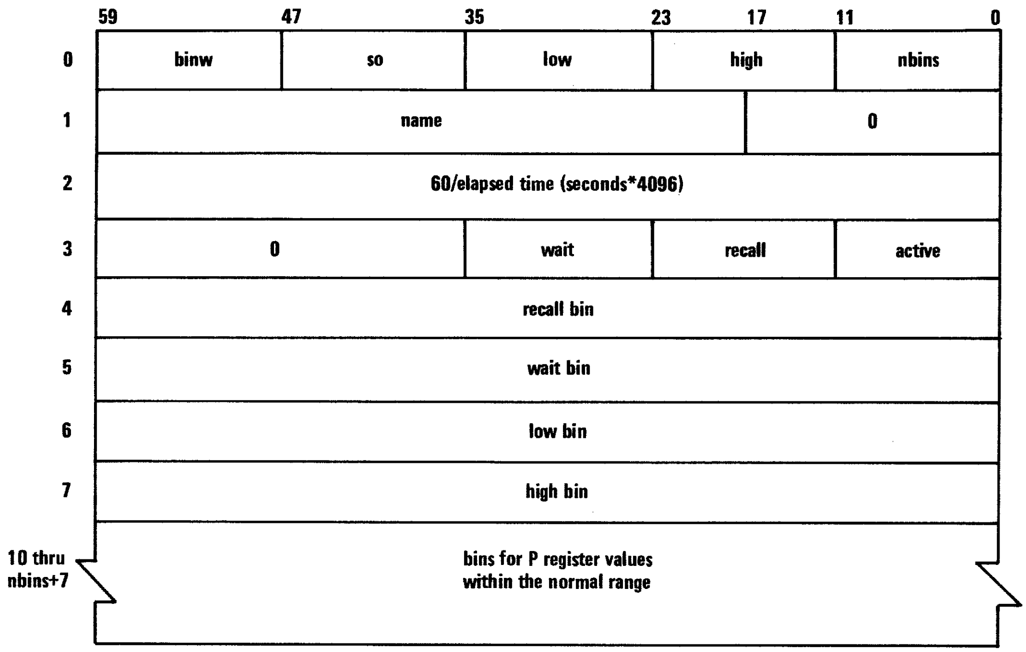
When SPY terminates, it writes the accumulated P register bins, special bins, and identification words on a file named DOSSIER. A program named PRNTSPY is provided that reads the data off the file DOSSIER and translates it into an easily understood report format, then writes it onto the OUTPUT file.

If spying is terminated by an abnormal program termination, DOSSIER may not be written because the operating system will not open a new file while an error flag is set. If the control card REWIND(DOSSIER) is placed in the job ahead of the point at which SPY is called, the file is allocated a small space on the disk and the file DOSSIER is written even though an error flag has been set at the control point.

†34 microseconds based on 1000-nanosecond peripheral processor cycle time and assuming no central memory conflicts.

DOSSIER File Format

The file DOSSIER produced by SPY has the following format:



If so=0, 1, or 2, the elapsed time is CPU time used. For so=10B or 20B, the elapsed time is wall clock time and includes time during which the SPY program may have been swapped out.

The wait, recall, and active numbers in word 3 are the number of cycles that are used in the minimum loop during each of the program statuses.

Dual CPU

The SPY is called to run on a dual CPU mainframe, it automatically delegates the remaining portion of the job to CPU A. This has the same effect as if a CPA parameter were used on the job card. If a CPB parameter is used on the job card, SPY spies on CPU B instead of CPU A.

Swapping

SPY permits the job scheduler to swap out the job on which it is spying. The S.CPUSTS Status bit is set while SPY is running. SPY writes its collected data on a scratch file named DOSTIER and places itself in the delay stack. Peripheral processor programs in the delay stack are captured by the swapper and saved on the swap file. When the job is swapped back in, SPY restores its data from the scratch file DOSTIER and resumes spying. The S.CPUSTW bit is cleared by SPY is assure that the job will not be run before SPY is ready to start collecting data on it again.

Checkpoint

The checkpoint capability is not compatiable with SPY. If a checkpoint is attempted while SPY is running, SPY writes the file DOSSIER and stops spying.

Calling SPY

The central processor routine CPSPY$ contains three entry points that can be used to start spying: SPYONF, SPYCLL, and CPSPY.

SPYONF is provided for FORTRAN 4 Extended or FORTRAN 5 programs. The calling sequence is:

```
CALL SPYONF(low,high,name,binw)
```

SPYCLL is provided for COMPASS programs. The calling sequence is:

```
RJ    =XPYCLL
```

When this call is used, the parameters are passed in registers:

```
X1 = low
X2 = high
X3 = name
X4 = binw
```

CPSPY is a control-card-callable entry point. The control card format is:

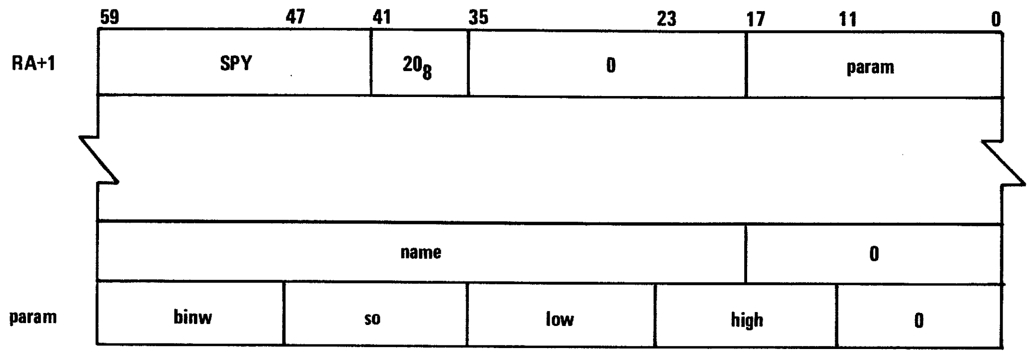
```
CPSPY(name,low,high,binw)
```

The CPSPY control card is used to spy on the control card that immediately follows it, usually an LGO. This permits spying on a program without modifying it.

SPY aborts the calling program without collecting values if it is called to run in a system assembled with IP.XJ≤0 (EXN systems).

SPY Calling Sequence

SPY is initited by an RA+1 call of the format:



name — The program name, left-justified with zero fill, that is used in the dayfile message SPYING ON name. It is also written on the file DOSSIER for PRNTSPY to include in the spying report.

binw — The bin width. It can have the values 1, 2, 4, 10B, 20B, 40B, or 100B.

low — The lower limit/100B.

high — The upper limit/100B.

so — The special option code:

- 0 = The value used by the SPYONF and SPYCLL entry points. Spying is initiated on the program that made the call. When initiated using so=0, spying can be terminated before the termination of the calling program. This is done by setting the word at param to zero and then using param as the operand of a RECALL statement with the auto-recall bit set.

- 1 = The value used by the CPSPY control card. When SPY is called with this value, it drops the central processor and calls 1AJ, which causes the next control card to be processed. SPY spies on the next control card processing until it is terminated by an END in RA+1 or by an error flag. The control card called by 1AJ must be one that uses the central processor.

- 2 = A special option, which is the same as so=0, except that spying cannot be terminated prior to the termination of the CPU program. SPY processing is more efficient when the special option is not zero because it is not necessary to watch the parameter word for SPY termination. The so=2 option is required if the parameter word is in an area that is later overlayed or used as buffer space.

  No entry points are provided for calling SPY with so=2.

10B = A special option for spying on monitor mode system programs. Spying will be performed from the time that SPY is called until the calling program terminates, except for time periods during which the calling program is in a swapped-out status.

This option only works correctly on systems assembled with IP.SPT≠0.

20B = A special option for spying on the user mode system programs, which are the job scheduler and storage move. Its processing is the same as so=10B.

If SPY is called with so=10B or 20B on a dual CPU mainframe, CPU B is temporarily turned off while spying takes place.

Terminating SPY

SPY stops spying whenever the job on which it is spying issues an END or is terminated abnormally for any reason, such as ERROR MODE, OPERATOR DROP, ABORT, and so forth.

If spying is initiated by a call to SPYONF or SPYCLL, it can be terminated by a call to SPYOFF.

For FORTRAN 4 Extended or FORTRAN 5 programs, the calling sequence is:

```
CALL SPYOFF
```

For COMPASS programs, the calling sequence is:

```
RJ      =XSPYOFF
```

PRNTSPY

PRNTSPY is an independent central processor program that prints out the report prepared by SPY. It is called by the control card:

```
PRNTSPY (lfn)
```

It reads the report file DOSSIER and the file lfn containing the relocatable binary of the program under scrutiny. The file lfn is used to build a value of load addresses and print out the reports relative to the program.

If PRNTSPY is called without the lfn parameter, the addresses in the reports are relative to RA.

The file DOSSIER must be released by a RETURN(DOSSIER) card following the PRNTSPY card to avoid duplicate printouts when multiple SPY runs are made.

To call SPY from an absolute overlay program written in COMPASS, include the appropriate code from the deck, CPSPY, and turn SPY on and off at appropriate points. Programs consisting of more than one overlay should turn SPY off before loading an overlay, and on again, following completion of the overlay load. To match records on DOSSIER with the correct overlay, the name arguement in the SPYONF call should be of the form namexy, with no trailing blanks, where the overlay cards are of the form:

OVERLAY (name, x, y).

Additional information may be obtained from the program listing.

DOCK USE

DOCK is a FORTRAN Extended 4 source language utility for extracting listable internal maintenance specification information from a COMPILE file generated from the NOS/BE program library.

The control card directive is in the following form.

DOCK(p1,p2,p3,...,pn)

Definition:

Default, if parameter is not specified

Assumed, if parameter is specified, but not equivalenced

p	Description
I	Name of program source file (assumed to be an update COMPILE or source file not exceeding 90-column BCD characters). Default = SOURCE, assumed = COMPILE.
L	Name of file containing documentation list (cannot be the same name as I). Default = assumed = OUTPUT.
F	Up to 25 characters to be printed in the bottom left corner of each page of documentation. Default = INT,Folio = $INTERNAL DOCUMENTATION.$ EXT,Folio = $EXTERNAL DOCUMENTATION.$ Assumed = $I M S.$
INT	Internal; all internal, external, and overview documentation will be listed on file L.
EXT	External; external and overview documentation will be listed on file L. Default = INT.
OVR	Overview; only overview documentation will be listed on file L.
INDEX	At the end of each routine processed an index is printed, all symbols found in location field of EJECT, SPACE, TITLE, and TTL cards. Default = INDEX off.
NR	No rewind of input file (I parameter); default = rewind of INPUT.
NT	No table generation. Default = table generation.
NP	No propagation of page numbers across routine. Default = on.
TE	Documentation file, L, formattted for input into program TEXTJAB. Default = off.

Default parameter settings include the following.

DOCK(I=SOURCE,L=OUTPUT,F=$INTERNAL DOCUMENTATION. $.INT,NP)

Assumed parameter settings include the following.

DOCK(I=COMPILE,L=OUTPUT,F=$INTERNAL DOCUMENTATION. $,INT,NP)

The following dayfile messages are issued by DOCK.

FL TOO SHORT FOR DOCK. (REQUIRES 12K).

Not enough field length was allowed; current minimum field length is 12K (octal).

FILE NAME CONFLICT.

Input, I, and List, L, file names are the same.

MEMORY OVERFLOW IN BUILDING INDEX TABLE.

Not enough field length for index table; increase by 4K (octal).

EMPTY INPUT FILE. NO DOCUMENTATION PRODUCED.

Input file was empty.

INPUT FILE NAME IS ILLEGAL.
OUTPUT FILE NAME IS ILLEGAL.

Illegal character specified in file name.

FILE EQUIVALANCE MAY NOT BE 0.

A file parameter cannot be set to zero.

SIS63 USE

The following control statement is required for execution.

SIS63(lfn)

where lfn is the SIS file name.

CIA USE

CIA is a peripheral processor routine used to collect statistics on CPU execution. The data collected by CIA can be used to help measure system performance, as an aid in determining resource utilization, and to help in finding system bottlenecks. CIA collects data on the percentage of time the CPU spends in each of four modes and reports this data in dayfile messages. These messages offer a quick method for determining how much of the CPU's time is spent working on user jobs and how much is spent on system activities or idling.

Percentages

To produce percentage dayfile messages, CIA is called either from the operator console or from a central processor routine via RA+1 with a call of the following format:

59	41	35	23	17 0
CIA	0	count	code	0

count The number of samples to take and messages to issue; default is 30.

code The internal code, having the following values:

1 =	Issues a dayfile message and starts a new set of samples at approximately 8-second intervals	
2 =	Increases the interval by a factor of 8 to approximately 65 seconds	
3 =	Increases the interval again by a factor of 8 to approximately 8.7 minutes	
5 =	Same as 1, but for CPU B	Applies to dual CPU mainframes only
6 =	Same as 2, but for CPU B	
7 =	Same as 3, but for CPU B	

This class of CIA call only works correctly on systems assembled with IP.XJ>0.

All calls for percentages should be codes 1, 2, or 3. If there are two CPUs, CIA calls a second copy of CIA using the codes 5, 6, or 7 to measure the percentage utilization on CPU B. If CPU A has been turned off, CIA modifies itself to measure the percentages on CPU B.

Dayfile messages produced are of the following form:

CPx a Mmm.m Sss.s Uuu.u Iii.i

x =	A for CPU A, B for CPU B
a =	* for 8-second intervals ** for 65-second intervals *** for 8.7 minutes
mm.m =	Percentage of central processor time spent in monitor mode
ss.s =	Percentage spent on system user mode routines - scheduler and storage move
uu.u =	Percentage spent on user programs
ii.i =	Percentage spent idle

If CIA is called with a code 1, it remains at the control point from which it is called and the dayfile messages appear on the job dayfile as well as on the system dayfile.

If CIA is called with a code 2 or 3, it changes its control point assignment to control point zero. The dayfile messages only appear on the system dayfile. This leaves no activity at the control point that made the call so that it can be terminated immediately. The percentage reporting at control point zero continues until the count is completed or a system deadstart occurs.

CPMET USE

CPMET (CP Monitor Execution Timing) is a central processor routine that is called to produce a report on the activities of CPMTR. This report, which is primarily of interest to those making modifications to the operating system or to those analyzing its internal performance, describes the frequency and duration of the various tasks performed by CPMTR. Because the efficiency of CPMTR execution has a significant effect on system performance, and because the types of activities undertaken by CPMTR largely reflect the nature of overall system operation, the data included in this report can serve as an aid in assessing the effect of system modifications on overall performance and in analyzing system activity, discovering bottlenecks, and so forth. The data in the report is obtained through calls to CIA.

The report produced by CPMET (figure II-8-1) contains four basic sections: monitor functions, M.ICE subfunctions, RA+1 calls, and overall run statistics. The monitor function section contains the octal code and symbolic name for each CPMTR function executed during the run, the number of times it was called, and statistics concerning the time required to execute that function. Entries in this section for PPMTR codes reflect the normal conflicts that occur within the PPMTR-CPMTR linkage and division of labor. An entry named SYSPROG accounts for those times when CPMTR was initiated while user mode system programs were executing, either by the programs themselves or by a peripheral processor (PPU) routine requesting execution of a CPMTR function. This entry is arbitrarily numbered 100B on the report (77B is the highest numbered monitor function code).

An entry named RAPLUS1 records data on CPMTR executions for requests made through a program's RA+1. This entry is arbitrarily numbered 101B.

The second section of the CPMET report is a detailed breakdown (in the same format) of the M.ICE subfunctions.

CPU EXECUTION STATISTICS

FCN CODE	NAME	CALLS	TOTAL	AVERAGE	LONGEST	SHORTEST
			[----------	TIMES IN	MICROSECONDS	----------]
2	M.CLRST	1	202	202	202	202
5	M.RCLCP	10	1237	123	175	67
6	M.ICE	187	87805	469	1498	67
10	M.SLICE	20	1988	99	148	67
12	M.RCH	64	3262	50	67	40
100	SYSPROG	2	188	94	94	94
101	RAPLUS1	16	4636	289	931	94

FCN CODE	NAME	CALLS	TOTAL	AVERAGE	LONGEST	SHORTEST
			[----------	TIMES IN	MICROSECONDS	----------]
0	EX.CMSM	0	0	0		
1	EX.ECSM	0	0	0		
2	EX.PLI3	62	24917	401	1093	67
3	EX.SPM	6	2940	490	742	256
4	EX.SS	0	0	0		
5	EX.SCH	2	215	107	121	94
6	EX.SCH1	0	0	0		
7	EX.REQEB	0	0	0		
10	EX.RELEB	0	0	0		
11	EX.CBM	0	0	0		
12	EX.SPRCL	47	16946	360	1039	148
13	EX.STAT	48	37344	778	1498	229
14	EX.NXTPB	22	5443	247	391	175
15	EX.FLHB	0	0	0		
16	EX.CSWAP	0	0	0		
17	EX.AUTEB	0	0	0		
20	EX.ECD	0	0	0		
21	EX.ECR	0	0	0		
22	EX.ECW	0	0	0		
23	EX.CEM	0	0	0		
24		0	0	0		
25	EX.ECLDV	0	0	0		
26	EX.BKSPF	0	0	0		
27	EX.LNKON	0	0	0		
30	EX.LNKIN	0	0	0		
31	EX.BOOT	0	0	0		
32	EX.TAT	0	0	0		
33	EX.RBT	0	0	0		
34	EX.SSF	0	0	0		

Figure II-8-1. CP Monitor Execution Timing Report (Sheet 1 of 2)

RA + 1 CALLS

NAME	CALLS	TOTAL	AVERAGE	LONGEST	SHORTEST
		[----------	TIMES IN	MICROSECONDS	----------]
EPFP	6	834	139	229	94
RCL:	5	794	158	202	94
CIOP	5	3008	601	931	499

CIA CALLED	1	TIMES
FIRST START	13.09.11.	
LAST STOP	13.09.12.	

UNLISTED RA+1 CALLS	0
TOTAL MONITOR CNTS	3534
TOTAL USER CNTS	33511
MONITOR MILLISECS	99
USER MILLISECS	938
TOTAL MILLISECS	1037
MONITOR PERCENT	9.57
USER PERCENT	90.42
RUN MILLISECONDS	1083

Figure II-8-1. CP Monitor Execution Timing Report (Sheet 2 of 2)

The third section is similarly a detailed breakdown of the RA+1 calls reported in the RAPLUS1 entry in the first section. The number of calls and timing statistics in this section are the same as in the preceding sections.

Timing information in this section is for CPMTR processing of the request only; time used by the PPU program requested (if any) is not included.

The name entries in this section are the display code equivalent of the high-order 24 bits of the RA+1 request. The first three characters are (generally) the name of the PPU routine requested. The fourth character indicates which bits between bits 36 and 41, inclusive, were set in the call. Most calls have one of the following fourth characters:

P = Bit 40 set - a request made with automatic recall

: = No bits set

The last section of the report supplies statistics on the overall CPMET execution. It includes the following data:

CIA called	Number of calls to peripheral processor routine CIA to collect raw data.
First start Last stop	Time-of-day values for the first and last samples supplied to CPMET during this run.

Unlisted RA+1 calls	Number of RA+1 calls that could not be recorded by name in the third section because of internal CPMET table overflow; normally is zero.
Total monitor counts Total user counts	Internal CIA count of cycles in monitor/user mode.
Monitor milliseconds User milliseconds Total milliseconds	Time during the CIA data collection phase that the CPU was executing monitor/user mode routines.
Monitor percent User percent	Monitor - MS./Total MS. *100 User - MS./Total MS. *100
Run milliseconds	Total time from the first CIA sample to the last.

All time values reported by CPMET except run milliseconds, first start, and last stop are derived from CIA internal cycle counts. Therefore, if CPMET/CIA is run on a CYBER 170 having a system assembled with IP.PPS2X=2, these values should be divided by 2.

CPMET may be called by control card or from the operator console. The following calls are accepted:

CPMET.	Call CPMET to collect data and write a report to the file named OUTPUT.
CPMET,lfn.	Call CPMET to collect data and write a report to the file named by lfn.

The number of times that CPMET calls CIA to collect data is determined by the value of sense switch 1. If sense switch 1 is zero (default), CIA is called once. Otherwise, the number of calls is determined by an assembly constant in CPMET (value of symbol N.CALLS; currently 500). Data collection can be stopped after any call to CIA by entering n.OFFSW1 from the operator console.

Calls issued by CPMET to CIA are only processed correctly on system assembled with IP.SPT≠0.

FORTRAN EXTENDED VERSION 4 9

RELEASE MATERIALS

FORTRAN Extended 4 is released on one reel of tape, PL7, which contains the compiler. PL7A is used by those installations who have purchased the single-pass time-sharing (TS) version of FORTRAN Extended. The installation of FCL 4 (PL8) mathematical and I/O libraries is required for FORTRAN Extended 4 execution (refer to section 34).

The structure of PL7 or PL7A is:

File 1	Program library of the FORTRAN Extended 4 compiler
File 2	Relocatable binary
File 3	Absolute overlay binary

LIMITATIONS

Install all applicable Integer Multiply FCOs. All code generated by the compiler assumes the existence of the Integer Multiply.

If FORTRAN Extended 4 is installed on a model 71, 72, 73, 171, 172, 173, 174, 720, or 730 with the MODEL installation parameter (in IPARAMS) correspondingly set, the object code produced will execute properly but will not be optimal for a model 74, 175, 750, 760. If MODEL is set to 74 or 175, the object code produced will execute properly on a model 71, 72, 73, 74, 171, 172, 173, 174, 175, 720, 730, 750, 760, 865, 875, or 180-class mainframe but will be optimal only for the model selected. On models other than 180-class mainframes if the MODEL parameter is set to 176, the compiled object code will not execute correctly on other models when the source programs contain LEVEL 2 (direct access LCM) statements, but will execute correctly although not optimally on other models when the source programs do not contain LEVEL 2 statements.

Object code compiled by FORTRAN Extended 4 on a model 71, 72, 73, 74, 171, 172, 173, 174, 720, or 730, cannot be executed on a model 76 running under the SCOPE 2 operating system. On the lower CYBER models, object code consists of one logical record for each program unit, whereas the SCOPE 2 operating system loader accepts only W records.

When the FTN control statement specifies either the C or E option, the compiled object code is produced as symbolic COMPASS source language, rather than executable binary. The C and E options may not be selected in TS mode.

COMPILER INSTALLATION PARAMETERS

The FORTRAN Extended 4 compiler program library, PL7 or PL7A, is distributed with installation parameters properly set for normal installation on any CYBER 70 or CYBER 170 machine. It should be noted that the system text IPTEXT should contain parameter values which are consistent with the CYBER model on which the compiler is installed and executed.

To ensure correct code generation, set the MODEL micro in deck IPARAMS on PL1A to the correct value for the target machine.

The installation options are located in the common deck OPTIONS and deck FTN. OPTIONS is called by TSTEXT, FTNMAC, and FTNTEXT; because of its global nature, reinstall the compiler whenever parameters are changed. Installation parameters in FTN may be revised through a standard maintenance run (installation deck PL7I).

You may obtain current UPDATE sequence numbers for installation options by assembling FTNMAC (or TSTEXT, FTNTEXT; the FTNMAC listing is much shorter) and/or FTN, depending on the parameters of interest. FTN contains the installation parameters for default control statement settings, control statement error processing, default file names, object-time input/output buffer length, and compiler overlay library names. The remaining parameters are in OPTIONS (FTNMAC/FTNTEXT/TSTEXT).

The default external and internal file names used by the compiler include the following.

INPUT	Source program input
OUTPUT	Compiler listable output
LGO	Relocatable binary object code
COMPS	COMPASS formatted symbolic object code (E option only)
ZZZZZFC	Internal symbolic object code (E option not selected)
ZZZZZRL	Internal intermediate language
ZZZZZRM	Internal reference map
ZZZZZOP	Internal OPT=2 and DEBUG-mode random file

All files are formatted according to suitable operating system standards. File formats cannot be changed through FILE control statements. (CYBER Record Manager has not been implemented for compile-time I/O. The upper CYBER implementation of Record Manager has been designed only for standard file formats; results are unpredictable if FILE statements are used.)

COMPILER PROGRAM LIBRARY STRUCTURE

When a full UPDATE is performed on PL7 or PL7A, the following nine records are written on the compile file.

Contents	Overlay	Program Library Deck Names
1. TS mode global assembly text (used only for installation or maintenance)		TSTEXT
2. Object code macro definition text		FTNMAC
3. Two-pass global assembly text (used only during installation or maintenance)		FTNTEXT

	Contents	Overlay	Program Library Deck Names
4.	Master controller	(0,0)	FTN
5.	TS option one-pass compiler. This record is empty if the compile file was produced from PL7, the non-time-sharing compiler.	(1,0)	TABLES-LIST
6.	Two-pass compiler (OPT=0, 1, or 2) batch compilation controller	(2,0)	LSTPRO
	Error message text expander	(2,3)	FTNMSG
	Pass 1 (non-DEBUG)	(2,1)	PS1CTL-PH1CTL
	Reference map processor and assembler	(2,5)	PS3CTL-REFMAP
	Pass 2	(2,2)	CLOSE2-MACROX
7.	FRAME (non-DEBUG) COPYL skeleton decks		FRAME
8.	DEBUG FRAME COPYL skeleton decks		FRAMDBUG
9.	Pass 1 DEBUG code	(2,4)	DBGPHCT-SAVREGS

TSTEXT

TSTEXT is a global set of macro, micro, and symbol definitions needed to assemble the (1,0) overlay. When the compiler is installed and maintained, TSTEXT is first assembled as a local text file and then accessed by the COMPASS or FORTRAN Extended 4 G option parameter.

FTNMAC

FTNMAC is a system text that contains macro defini- tions needed to assemble symbolic object code compiled by FORTRAN Extended 4. Normally, FORTRAN Extended 4 produces executable binary (not symbolic) object code without using FTNMAC. However, if the C or E option is selected, FORTRAN Extended 4 produces object code in a symbolic form for COMPASS assembly. The symbolic code contains many macro calls which must be externally defined for successful assembly. The macro definitions are available in FTNMAC, which is assembled and added to the operating system nucleus library during compiler installation.

FTNMAC must be assembled with the same version of COMPASS as is present when FORTRAN Extended 4 is eventually added to the running system.

FTNTEXT

FTNTEXT is a global set of macro, micro, and symbol definitions needed to assemble the (0,0) and (2,n) overlays of FORTRAN Extended 4. When the compiler is installed and maintained, FTNTEXT is first assembled as a local text file, and then is accessed by the COMPASS or FORTRAN Extended 4 G option parameter.

The (0,0) overlay is the master controller that does the following.

- Scans, validates, and stores FORTRAN Extended 4 control statement option parameters.
- Initializes the compiler according to control card options and available memory.
- Loads all compiler primary and secondary overlays.
- Processes all operating system action requests.
- Loads and communicates between the COMPASS (1,0) overlay and FORTRAN Extended 4 for intermixed COMPASS language program units.

TS MODE OVERLAY

The (1,0) overlay contains the entire TS mode compiler (except for the (0,0) controller) and remains resident in core for the entire compilation. Source input is read, listed (if requested), and used to generate code. If COMPASS subprograms occur on the input file, COMPASS is loaded to assemble them and FTN10 is reloaded, if necessary for subsequent FORTRAN subprograms.

OPTIMIZING AND DEBUGGING MODE OVERLAYS

(2,0) Overlay (FTN20)

This overlay is the batch controller for compiling multiple FORTRAN program units. It contains the symbol/label table lookup subroutines; the central compiler input/output subroutines; the batch compilation reinitialization code; miscellaneous utility subroutines; and the compiler malfunction report package.

(2,3) Overlay (FTN23)

This overlay expands two-word error table entries into full line error messages. Load only if errors were detected during Pass 1 of a compilation.

(2,1) Overlay (FTN21)

This overlay is the first pass of the compiler under normal mode (when the DEBUG option is not selected by the control statement D parameter). It performs a lexical, syntactic, and semantic analysis of each FORTRAN program unit. Source language input is translated through a lexical element language (E-list) to a register-independent internal language (R-list). Source language errors are detected and saved in an error table for subsequent expansion. Intermixed COMPASS language programs are recognized and either copied to an internal file or transmitted directly to COMPASS, depending on control card option selection.

(2,2) Overlay (FTN22)

This overlay is the second pass of the compiler. It optimizes and generates symbolic object code from the R-list produced by Pass 1.

(2,5) Overlay (FTN25)

This overlay is the third pass of the compiler. A reference map is produced, if requested. The symbolic code is then assembled as executable binary object code, either by a fast one-pass internal assembler or (if the C option is selected) by the slower COMPASS assembler. This overlay can be combined with the (2, 2) overlay during installation, as selected by the symbol .OVL in OPTIONS. Compiler loading time is reduced, but you must increase compiler field length.

(2,4) Overlay (FTN24)

This overlay is loaded only when the DEBUG option is selected on the FTN control statement. It is loaded instead of the normal Pass 1; it contains, in addi- tion to all normal Pass 1 code, processing subroutines for DEBUG statements. The overlay is formed during installation by assembling the DEBUG subroutines and then replicating the normal Pass 1 code with the COPYL utility.

Minimum DEBUG field length is 63K (octal) or approximately OPT=0+15K.

INSTALLATION INSTRUCTIONS

The compiler installation decks provide a method for introducing the FORTRAN Extended 4 compiler into a NOS/BE system. The first job PL7I updates the program library, producing a new program library tape including supplemental binary files. Run deck PL7E following PL7I but before attempting installation of the object library when the running system modification approach to building systems is used.

Deck PL7I references IPTEXT and CPUTEXT; part III of this document contains a cross-reference map of referencing routines versus IPARAMS symbols. Deck PL7I also requires access to the COMPASS program library to access several common COMDECKS.

You can obtain compiler installation job decks PL7I and PL7E, and verification program PL7V from the Installation Deck program library using the procedure outlined in part I, section 1.

Decks PL7E and DST3 need not be run if the user library approach is being followed.

COBOL VERSION 4.7 10

SECTION 10. COBOL VERSION 4.7 HAS BEEN REMOVED.

RELEASE MATERIALS

Sort/Merge 5 is released on release tape PL78. PL78 contains the following files:

File	Contents
1	Sort/Merge 5 program library
2	Sort/Merge 5 (0,0) overlays
3	SRT5LIB
4	Relocatable binaries for capsules

HARDWARE CONFIGURATION

Sort/Merge 5 requires the same minimum hardware configuration as NOS/BE.

INSTALLATION PARAMETERS

Sort/Merge 5 has no installation parameters.

INSTALLATION PROCEDURE

You may obtain job decks PL78I, PL78E, and PL78V from the Installation Deck program library using the procedure outlined in part I, section 1 of this document.

The installation jobs function as follows:

PL78I Updates the program library with modifications producing a new program library tape including assembled binary information as supplemental files. This job allows creation of a revised release tape.

PL78E Adds Sort/Merge 5 to the running system or user libraries. PL78E can use either the released PL78 or a tape created by job PL78I as input.

INTERCOM VERSION 4 12

SECTION 12. INTERCOM VERSION 4 HAS BEEN REMOVED.

ALGOL 60 VERSION 4 13

SECTION 13. ALGOL 60 VERSION 4 HAS BEEN REMOVED.

REMOTE DIAGNOSTIC FACILITY 14

RELEASE MATERIALS

The Remote Diagnostic Facility (RDF) should be installed only on 180-class mainframes and is released on one reel of magnetic tape (PL14) containing the INTERCOM 5 and RDF program libraries. Installations not licensed for INTERCOM 5 should install PL14A, which contains only the programs necessary to install RDF.

INSTALLATION PARAMETERS

Refer to section 28 for further information about the use of the IP.IND parameter.

EQUIPMENT STATUS TABLE

The equipment status table (EST) is established when the PL1A1 deck is used to install NOS/BE. It must contain an entry for the two-port mux (TPM) to be used by RDF. The EST macro has the following format:

```
RM,CH=ch,EQP=eqp,MOD=OFF
```

Parameter	Description
CH	Octal channel number for the TPM. If you omit this parameter, 15_8 is assumed.
EQP	Octal equipment number for the TPM. If you omit this parameter, 15_8 is assumed.
MOD	Entering OFF turns the entry off, if desired (optional).

Refer to Equipment Configuration in part II, Section 1, for further information.

INSTALLATION PROCEDURES

Install RDF from PL14 or PL14A. RDF is included with PL14; if your site does not have PL14, install PL14A. Create the PL14A installation job by defining the NOINT deck option with the PL14 installation job (PL14I). (Refer to section I-1 for more information on the NOINT option.)

Assemble CMR with an RM entry in the EST.

Generate a deadstart tape with the new CMR.

RDF USE OF PASSWRD

Establish a PASSWORDS permanent file to allow users to login to RDF. Refer to PASSWRD (Section 28) for further information.

RELEASE DESCRIPTION

The CYBER Cross System executes under NOS/BE to provide support for the CYBER 18 minicomputer and the 2550 series of host communication processors. The CYBER Cross System is composed of the following.

Component	Implementation Language
Pascal Compiler	Pascal
Format Program	FORTRAN Extended 4
Pascal Cross-reference Program	Pascal
Macro Assembler	COMPASS
Macro File	Assembly
KRONTXT	COMPASS
Micro Assembler	FORTRAN Extended 4
Library Maintenance Program	FORTRAN Extended 4
Link Editor	Pascal

The CYBER Cross System supports the generation of load modules which may be executed on a CYBER 18 minicomputer or a 2550 communications processor.

HARDWARE CONFIGURATION

The CYBER Cross System requires a minimum of 125000 octal words of central memory for installation and execution.

RELEASE MATERIALS

CDC CYBER Cross System is released on release tape PL50, the structure of which follows.

File Number	Record Number	File Content		File Type
1	1	Update Program Library		PL
2	1	Format Program	(FRMT)	ABS
2	2	KRONTXT	(KRONTXT)	OVL
2	3	Macro Assembler	(ASSEM)	OVL
2	4	Macro File	(SMAC17)	OVL
2	5	Micro Assembler	(MASSEM)	ABS
2	6	Library Maintenance Program	(MPLIB)	ABS
3	-	(empty)	-	-
4	1	Pascal Compiler (standard)	(PASCAL)	OVL
5	1	Pascal Cross Reference Program	(PASXREF)	DATA
6	1	Link Editor	(MPLINK)	OVL
7	1	Edit	(MPEDIT)	OVL
8	1	Pascal Compiler (CCP)	(PASCAL)	OVL
9	1	6000 Pascal Compiler	(PASBN01)	OVL

File 9 contains the binary of the 6000 Pascal Compiler, required for compiling the PSCAL compiler and the Link Editor, and for compiling and executing the Pascal cross-reference program.

PASCAL ORGANIZATION

The Pascal compiler is organized in a file structure; the components of the compiler are records on the file. The first record of the file acts as a main overlay program and controls loading and execution of the other records. Because of this file structure, the Pascal compiler cannot be entered into a library via EDITLIB, but must be cataloged as a permanent file (PASCAL). The Pascal file structure follows.

Record	Name	Function	Implementation Language
1	POSYS	Controls processing	COMPASS
2,3	PASCAL	Compiles programs	Pascal
4	SYMIO	Performs disk I/O	COMPASS
5	ERRMSS	Table of error messages	Text
6,7	PASDMP	Prints object code listing	Pascal

PASCAL COMPILER SYMBOL TABLE

The number of entries in the in-core symbol table in the standard version of the Pascal compiler is 1792. The maximum number of global symbol definitions is 1536. The compiler has an execution field length of 77000 octal CM words. For compilation of programs which have more than 1536 global symbols, such as CCI or CCP. Increase the maximum number of globals (this does not increase the execution field length).

To generate a Pascal compiler for installing CCI or CCP, the maximum number of global symbols is increased to 6144. Generation of a compiler for CCI is accommodated as a CYBER Cross System installation option (refer to Installation Procedure).

INSTALLATION PROCEDURE

You may obtain job decks PL50I, PL50C, and PL50V from the installation deck program library using the procedure outlined in part I, section 1 of this document.

The installation jobs function as follows.

PL501 — Updates the program library with modifications to produce a new program library tape including binary files. If PL50I is extracted from the installation deck program library with NOCCP not defined (refer to part I, section 1), a 125K version of the Pascal compiler is produced and written on file 8 of the new PL50. The PAGSIZE modification, however, is not included on the new program library file (file 1) of PL50. If job PL50I is extracted with NOCCP defined, the 77K version of the Pascal compiler is produced and written on file 4 of the new PL50. PL50I requires a field length of 135000 octal words to compile the 125K PASCAL and 77000 octal words to compile the 77K version. Defining CATALOG causes job PL50I to catalog the new PL50 binaries as permanent files from which they may be executed. The CATALOG option is sensitive to the NOCCP symbol so that if NOCCP is not defined, a 125K PASCAL is cataloged; if NOCCP is defined, the 77K version is cataloged.

PL50C Catalogs the CYBER Cross System binaries from PL50 as permanent files from which they may be executed. Not defining NOCCP causes the 125K PASCAL compiler to be cataloged. Defining NOCCP catalogs a 77K version of PASCAL. If PL50I is run with CATALOG defined, PL50C is not required.

Because the PL50 installation jobs do not enter program binaries into libraries, the DSTn jobs are not applicable.

PL50V Verifies installation of the CYBER Cross System. It uses the permanent files created either by job PL50I with CATALOG defined, or by job PL50C.

RELEASE DESCRIPTION

The CYBER Loader runs on CYBER 170, CYBER 70, and 6000 Computer Systems. CYBER Loader runs under NOS/BE and requires the same minimum hardware as NOS/BE 1.

RELEASE MATERIALS

The release tape for CYBER Loader is PL1E which contains a source program library as file 1 and the assembled binary as file 2.

GENERAL DESCRIPTION

PL1E contains LDRTEXT and programs PILOAD, LOADER, LOADU, UCLOAD, LDRCNTL, SEGBILD, FDL.RES, FDL.OCR, FDL.MMI, FOL.RES, SEGRES, TRAP and TRAPPER together with their associated common decks and higher level overlays.

This essentially comprises what is commonly known as the Control-Card-callable Basic Loader, User Call Loader, Fast Dynamic Loader, Overlay Loader, Segment Loader, Fast Overlay Loader, Loader Control Card Processor, and the Debug Aids Package. For further common deck and overlay structure information, consult the Loader Reference Manual or the IMS.

Note that the program library (PL1E) for the CYBER Loader contains no Peripheral Processor (PP) programs. These routines are resident on the system program library PL1B. These routines must exist in the system for correct loader function. The primary loader interface PP programs include the following.

PP Program	General Function
LDL	Read one library directories, loader control word, set protect bit, assign library files.
LDV(LDW)	Perform physical loading

NOTE

LDV may call PP program LDW depending on type of load function

INSTALLATION PARAMETERS

CYBER Loader obtains installation parameters from its own local LDRCOM deck. The following installation parameters for the control statement initiated loader may be set at LDRCOM.13 in the update of PL1E. The values shown in parentheses are default values.

IP.PSET (11B)

Core presetting options include the following.

0	Same as 1					
1	Preset to	0000	0000	0000	0000	0000B
2	Preset to	7777	7777	7777	7777	7777B
3	Preset to	1777	0000	0000	0000	0000B
4	Preset to	3777	0000	0000	0000	0000B
5	Preset to	6000	0000	0000	0000	0000B
6	Preset to	4000	0000	0000	00	addr
7	Preset to	2525	2525	2525	2525	2525B
10	Preset to	5252	5252	5252	5252	5252B
11	Preset to	6000	0000	0004	0040	0000B+*

For (6) each location contains its address in the lower 18 bits.

IP.REW (1)

If one, the load file is rewound prior to beginning to load. If zero, no rewind takes place.

IP.LDBG (0)

If nonzero, conditional code to aid in debugging the Loader is assembled. Additional information is available in the Loader IMS.

IP.LDER (1)

Error processing by the loader may be one of the following.

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

IP.FLINC (4000B)

Amount by which field length is increased if loader needs more field length for table construction. May vary up from 100B in increments of 100B.

IP.FLMSG (0)

If nonzero, a dayfile message giving the FL required for loading and execution will be issued for relocatable loads when there is no map.

IP.LRT (0)

If nonzero, a dayfile message is issued giving various time and memory measurements. If IP.LRT≥1000B, then the value (IP.LRT-1000B) is placed in bits 35-24 of the MSG call. For example, a value of 1003B will place the message in the system dayfile, the job dayfile, and will also be seen at an interactive terminal.

IP.MAP (3)

Default Loader MAP options include the following.

0 =	MAP(OFF)	No Map
3 =	MAP(PART)	S,B options
13B =	MAP(ON)	S,B,X options
17B =	MAP(FULL)	S,B,E,X options

S	Loader statistics and error messages only
B	Block names, addresses and lengths
E	Entry point list
X	Cross reference list of external references

INSTALLATION PROCEDURES

Installation of the CYBER Loader requires that job decks PL1EI and PL1EE be obtained from the installation deck program as outlined in part I, section 1.

PL1EI is a maintenance deck which can be used to create a revised program library and binary file. PL1EE can be used to enter CYBER Loader into the running system or user libraries from either the released PL1E or a tape created by PL1EI. After deck PL1EE has completed, run job DST3 to create a deadstart tape of the running system. Job decks PL1EE and DST3 need not be run if the user library installation process is being followed.

844-21 and 844-41 Factory Format Support is a software feature that provides FORMAT/FDP and is applicable to NOS/BE running on CYBER 170, CYBER 70, and 6000 Computer Systems that include CEAIDS/D44 and level A06 (or above)OSY controlware. A08 (or above)OSY controlware is required for 844-41 disk units.

RELEASE MATERIALS

The release tape for FORMAT/FDP is PL1F which contains a source program library as file 1 and the assembled binary as file 2.

GENERAL AND OPERATIONAL DESCRIPTION

PL1F contains the FORMAT/FDP utility which enables on-line support of 881 and 883 factory formatted disk packs. Factory formatting is the process of preparing a disk surface for use by recording addresses on the disk surface for cylinders, tracks, and sectors. All 881 and 883 disk packs are factory-formatted, surface tested for flaws, and certified for use before shipment to the customer. Under normal conditions, these disk packs remain in use with few further problems. 844-21 and 844-41 Factory Format Support is intended to aid in maintaining disk packs in a usable state and correcting problems that might be encountered.

It is strongly recommended that you designate one person as disk pack coordinator, responsible for maintaining all installation disk packs. Only the disk pack coordinator should be allowed to use the pack formatting procedure and the FORMAT utility to do all disk pack formatting. Operators should be forbidden to attempt the operations described herein.

One master aligned disk drive is required for formatting all 881 and 883 packs. As the FORMAT utility tends to monopolize the controller (formatting is actually a controller function), disk pack formatting should be done as a hands-on activity and should not be allowed during a production environment. Operators should be instructed to drop any FORMAT job.

FLAWS

All 881 and 883 disk packs are fully surface-tested before being certified for use. Any flaws detected during disk surface analysis are recorded on the disk pack in the utility sector (located at cylinder 410D for 881, cylinder 822D for 883) track 0, sectors 1 and 2 (sector 0 contains the pack serial number).

Surface analysis detects two types of surface abnormalities, hard flaws and soft flaws. Hard flaws are small areas of the disk surface where data cannot be successfully read and written. Soft flaws are those small areas where doubt exists as to the accuracy of repeated reads and writes. To avoid use of these areas, the sector in which they occur is flawed (removed from use) by recording the address (cylinder/track/sector) in the utility sector. The operating system then reads the utility sector and flags known flawed areas as nonusable.

The disk pack coordinator must ensure that all installation disk packs appear flaw-free to the system; that is, all hard and soft flaws must be noted in the utility sector of the disk pack so that the flawed areas will not be accessed by the operating system.

When parity errors are encountered on a disk pack during customer usage, return the pack to the disk pack coordinator for the site, who can then run D44 tests to determine if new flaws have appeared since the last surface test and update the utility sector accordingly. D44 is described in the Concurrent Maintenance Library Reference Manual, which is only available to sites having a Control Data maintenance services agreement.

To keep track of the existing flaws on each disk pack, create and maintain a disk surface analysis record (DSAR) for each disk pack in use at the installation.

DISK PACK CONDITION

All disk packs in use should have intact factory format information, should have all discovered flaws recorded in the utility sector, and should encounter no parity errors during use.

Disk packs not in use may fall into one of the following categories.

- The factory format information is intact but the content of the utility sector is not accurate; parity errors (recovered or unrecovered) are experienced by the customer. The disk pack coordinator must run D44 tests, compare the output with the DSAR and update both the utility sector and the DSAR. The FORMAT utility, described later in this section, is the only present method for updating the utility sector. When this operation is complete, the pack may be returned to use.

- The factory format data was destroyed or never existed. Customer Engineering must use MSL/FMT utility to initialize factory format sectors. The disk pack coordinator can then run D44 tests. MSL/FMT is run only by Customer Engineering to initialize disk packs at the request of the disk pack coordinator.

- Disk packs with serial numbers below 819683 do not have soft flaws indicated in the utility sector. A disk surface analysis of these packs should be done using D44. Enter all flaws encountered in the utility sector using the FORMAT utility and noted on the DSAR. The packs may then be returned to use.

- New, repaired, or reconditioned packs should be verified using the FORMAT utility (V) and have a DSAR created before placing the pack in use.

NOTE

For both D44 and FORMAT utility operations, it is strongly recommended that you use only one drive to ensure drive-to-drive compatibility of the disk packs. The alignment of this master drive should be checked prior to any utility sector updates.

INSTALLATION INFORMATION

Installation deck PL1FI creates the binaries of FORMAT and FDP. This deck can start execution immediately after the successful installation of SCPTEXT in PL1AI. Deck PF1FE enters the assembled binaries produced by PL1FI into either the running system or the appropriate user libraries.

FORMAT UTILITY

The FORMAT utility package is intended solely for the purpose of maintaining 881 and 883 type disk packs for use with the 844-21/844-41 disk drives using 7054/7154/7155 controllers.

- The utility can retrieve the factory recorded manufacturing data, the factory recorded flaw data, and the utility flaw data from a factory formatted disk pack.
- Set/clear sector or track flaws on a factory formatted disk pack.
- Restore the address fields of a previously factory formatted disk pack. (This function is to be used only in the event of loss of addresses on the pack.)

CALL FORMAT AND PARAMETER OPTIONS

FORMAT is a control statement callable CP program that interfaces with a user/operator and a PPU program, FDP, to effect maintenance operations on an 881 or 883 type disk pack that has previously undergone factory formatting. The format of the call card follows.

```
FORMAT(p1,p2,...,pn)
```

The parameters are position-independent and may be any of the following.

I=lfn Defines the input file containing directives and data for controlling utility functions (default is INPUT).

L=lfn Defines the output file to receive information extracted from the disk pack, and so on. This is the standard output file (default is OUTPUT).

O=lfn Optional output file in addition to the file specified by L for information retrieved from the disk pack.

U=xxx Defines the EST ordinal (in octal) of the 844-21 or 844-41 on which the disk pack is mounted.

Unit must be logically OFF (to the system) and must not contain any active files.

The U parameter must be specified.

V Causes the utility to verify the address recorded on the disk pack. V is significant only when MODE = FETCH or MODE = RESTORE.

P=SN Declares the pack serial number of the pack to be processed. This is a decimal number that should exactly match the serial number recorded on the disk pack at the factory.

MODE= Declares the operational mode for the utility. Valid declarations follow.

ALTER	Indicates that the input file contains directives to control SET/CLEAR flaw operations.
FETCH	Indicates that the utility is to obtain the information contained on the factory sectors (CYL 410D, TRK 0, SEC 0 1,2 for 881 and CYL 822D, TRK 0, SEC 0,1,2 for 883) and copy it to the output file, and to the optional output file if specified.
RESTORE	Indicates that the utility is to restore addresses and flawed sectors/tracks per the utility flaw map. The utility flaw map must be intact or the program will abort.

Default parameters are equivalent to the following call.

```
FORMAT(I=INPUT,O=O,L=OUTPUT,MODE=FETCH,P=0,U=xxx)
```

You must declare the U parameter to initiate utility processing. The default SN(P=0) always produces an operator message (S/N MISMATCH) and requires a GO.

INPUT FORMATS

Input to FORMAT contains control directives and flaw data for updating the utility flaw map. Data contained on the input file is examined only when the operational mode has been declared as ALTER: the input file will not be accessed in either the FETCH or the RESTORE modes of operation.

Control directives follow.

SET	Declares the following data statements contain the addresses of flaws to be set and entered in the utility flaw map.
CLEAR	Declares the following data statements contain the addresses of flaws to be cleared and deleted from the utility flaw map.
FINIS	Declares the end of the input data. No information following this card will be processed by FORMAT. This directive is optional.

Data cards are of the following format.

```
x,cccc,tt,ss
```

x	S or T, to indicate a sector or a track flaw.
cccc	Octal number specifying the cylinder (0 - 632B for 881, and 0 - 1466B for 883).
tt	Octal number specifying the track (0 - 22B).
ss	Octal number specifying the sector (0 - 27B). ss field is ignored for track flaws.

All input data is checked to ensure that values are within range. Any errors in the input result in termination of the utility prior to accessing the disk. SET and CLEAR directives can be intermixed in the input; however, all CLEAR operations are performed before any SET operations. Any attempt to alter the status of the factory sectors results in an error. All control directives and data start in column 1. A maximum of 157 data statements can appear in the input stream.

OUTPUT FORMATS

Output generated by FORMAT always goes to the standard output file. Additionally, output generated as a result of a FETCH operation can be directed to a second file; this file can then be used as input to another program or disposed to either card or hard copy, and so on. Format of data in the optional output file is identical with input formats, however, no directive cards are used.

For all modes of operation, standard output contains the following information.

- Listing of the input stream, if any.
- Pack serial number and data of factory formatting from the manufacturing sector (CYL 410D, TRK 0, SEC 0).
- Listing of the factory flaw map as continued on CYL 410D, TRK 0, SEC 1 for 881; CYL 822D, TRK 0, Sec 1 for 883.
- Listing of the utility flaw map as contained on CYL 410D, TRK 0, SEC 2 for 881; CYL 822D, TRK 0, SEC 2 for 883.
- Listing of the utility flaw map following any changes resulting from SET and/or CLEAR directives. (MODE = ALTER only).
- Listing of flawed sectors and tracks as read from the disk during address verification (MODE = FETCH or MODE = RESTORE).

OPERATOR INTERVENTION AND CONSOLE MESSAGES

Operator intervention is required on all ALTER and RESTORE operations as a safeguard against accidental pack destruction. In addition, if the pack serial number parameter does not match the serial number recorded on the disk pack, the operator is given the option of dropping the job or overriding the condition and allowing the job to run.

The following console messages are displayed to inform the operator of the status of the function being performed or the need for intervention to continue processing.

ALTERING FLAW MAP S/N=xxxxxx; status message indicating utility flaw map is undergoing modification.

RESTORING ADDRESSES S/N=xxxxxx; status message indicating pack is currently undergoing restoration of address fields. Control point should not be dropped while message is displayed.

FETCHING FLAW DATA S/N=xxxxxx; status message indicating factory recorded data is being retrieved from CYL 410D, TRK 0, SEC 0,1,2 for 881; or from CYL 822D, TRK 0, SEC 0,1,2 for 883.

VERIFYING ADDRESSES S/N=xxxxxx; status message indicating read-only pass is being made across pack. Message is displayed after successfully fetching factory-recorded data and flaw maps or successfully restoring address fields if VERIFY option (V) was specified on program call card.

S/N MISMATCH - xxxxxx GO/DROP; flashed when P parameter is not identical to serial number found on pack. Operator must intervene to continue processing.

xxxxxx TO BE ALTERED GO/DROP; flashed whenever utility flaw map is to be modified. (MODE = ALTER). Operator must intervene to continue processing.

xxxxxx TO BE RESTORED GO/DROP; flashed whenever address fields are to be rewritten, (MODE = RESTORE). Operator must intervene to continue processing.

In all the preceding messages, xxxxxx signifies the serial number as read from the manufacturing data recorded in CYL 410D, TRK 0, SEC 0 for 881; and CYL 822D, TRK 0, SEC 0 for 883.

DAYFILE MESSAGES

In addition to the console messages, which are entered in the system and control point dayfiles, the following messages are entered in the dayfiles to record catastrophic conditions that caused the program to abort.

EST ORDINAL xxx INVALID OR UNAVAILABLE; indicates EST.ordinal xxx, defined by U=xxx on call card, is unusable. EST entry is printed in octal in output.

FILE EQUIVALENCE MAY NOT BE 0; indicates either input file or standard output file has been declared empty.

ILLEGAL FILE NAME - xxxxxx; indicates that file has been given an illegal name.

INVALID DATA IN INPUT STREAM; indicates that input file contains incorrect data. Refer to input stream listing for card in error.

INVALID PARAMETER ON PROGRAM CALL CARD; indicates at least one unrecognized or ill-formed parameter found.

MANUFACTURING DATA INVALID; indicates that one of the factory-recorded sectors containing either manufacturing or flaw data is either unreadable or not present. Refer to output for detailed status indicating actual problem. If factory-recorded data cannot be read, pack may not be processed using this utility.

SERIAL NUMBER MUST MATCH ON ALTER; indicates attempt made to modify utility flaw map without first obtaining exact match between P parameter and serial number recorded on the pack. Since this may result in destruction of valid data, override is disallowed. Refer to output listing for actual serial number read.

TABLE OVERFLOW ON INPUT; this message indicates that too many FLAW statements were found in the input stream. FLAW input limit is 157B flaws.

UNRECOVERABLE ERROR CONDITION OCCURRED; indicates utility operation was terminated because of nonrecoverable error. Refer to general and detailed status in output listing for specific error condition. If this condition occurs, it is likely that pack and/or drive is unusable in its present condition.

INSTALLATION PROCEDURES

You may obtain installation job decks PL1FI and PL1FE from the installation deck program library using the procedure outlined in part I, section 1.

PL1FI is a maintenance deck which can be used to create a revised program library and binary file. Job PL1FE can be used to enter FORMAT/FDP into the running system, after which job DST3 may be run to create a deadstart tape of the running system. If the user library installation process is being followed, PL1FE and DST3 need not be run.

CYBER DATABASE CONTROL SYSTEM 1.2

SECTION 18. CYBER DATABASE CONTROL SYSTEM 1.2 HAS BEEN REMOVED.

RELEASE MATERIALS

QUERY UPDATE is released on the program library tape PL55. The structure of the release tape is as follows:

```
File 1  Program library
File 2  QUERY UPDATE, and REPORT absolute overlays
File 3  Absolute binaries of QUERY UPDATE, and REPORT
File 4  Owncode linkage module; binary, relocatable format
```

HARDWARE CONFIGURATION

QUERY UPDATE requires the same minimum hardware configuration as NOS/BE. A minimum of 30K octal words of central memory is required to execute this product. A typical minimum job requires approximately 5K octal more for buffers in order to run.

INSTALLATION PARAMETERS

The common deck IPARAMS, present in system text IPTEXT, does not contain any parameter specific to QUERY UPDATE. IPARAMS is used to test for the installed character set (IP.CSET) and for the format of the system date (IP.YMD). Part III of this document contains a cross-reference map of QUERY UPDATE routines versus symbols in IPARAMS.

Assembly options are defined within QUERY UPDATE. At the time of release, the assembly options are set to values deemed most convenient or practical. For example, default report page size is 136 columns x 60 lines. To obtain an up-to-date listing of the assembly options, run a job containing the following control statements and directives (the program library for QUERY UPDATE should be available on file OLDPL).

```
UPDATE,Q,L=0.
COPYSBF,COMPILE,OUTPUT.
7/8/9
*IDENT CPT
*C TOPTION      SYMPL text containing the assembly options
```

NOTES AND CAUTIONS

QU 3 requires DDL 3 to run. DDL 3 is released on PL77. The installation deck expects to find the syntax table generator SYNGEN and UPDATE COMDECKS in PL77. Installation of QU 3 requires defining DMGMNT in jobs DST1, DST2, and DST3.

INSTALLATION REQUIREMENTS

1. Before installing QUERY UPDATE, install the following products.

CYBER Record Manager (AAM)	Including the AAM indexed sequential access method (IS), the direct access method (DA), the actual key access method (AK), the multiple index processor (MIP). The access method IS is mandatory; DA/AK/MIP are optional if the corresponding features are not used.
SYMPL Version 1	
CYBER Database Control System 2 (CDCS 2)	If CDCS is to used.
Sort/Merge Version 5	If the directive SORT is to be used. If Sort, CDCS, or one of the access methods are not installed, missing entry points show in the load map when Query Update is installed.
DBU Version 1	If logging is to be used.

2. In addition, SYNGEN, a special syntax table generator, is needed to compile Query Update. The installation deck expects SYNGEN to be found in PL77 as part of file 1.

3. The installation deck accesses the DDL 3 PL found in PL77 in order to obtain UPDATE COMDECKS for installing Query Update.

INSTALLATION PROCEDURE

You may obtain installation job decks PL55I, PL55E, PL55O and PL55V from the installation deck program library using the procedure outlined in part I, section 1.

PL55I would be used to modify the PL, build a newpl, assemble and compile the entire Query Update PL, and generate and save the relocatable and absolute binaries on the newpl.

Use PL55E to install Query Update into the running system or user libraries.

PL55O, applicable only in the user library method of installation, allows regeneration and replacement of absolute overlays in the user library plus creation of a new PL55 tape.

The main overlays of Query Update and REPORT are placed in the library NUCLEUS, the other overlays in the library SYSOVL. The Data Base Procedure linkage module and OVCAPSs are DMSLIB.

Query Update is able to produce reports using the DESCRIBE directive even if DDL Version 3 is not installed. Without DDL Version 3, the USE, EXTRACT, and other file manipulation directives cannot be processed. However, the DDL 3 PL is required in order to install Query Update.

DATA DESCRIPTION LANGUAGE VERSION 2.2 20

SECTION 20. DATA DESCRIPTION LANGUAGE VERSION 2.2 HAS BEEN REMOVED.

RELEASE MATERIALS

BASIC 3 is released on tape PL57. PL57 contains three files. File 1 contains the source code for BASIC 3 in Update program library format; file 2 contains the absolute binary of the compiler; file 3 contains the relocatable binaries of the runtime library routines.

HARDWARE CONFIGURATION

The minimum configuration to operate BASIC 3 in batch mode is the same as the NOS/BE minimal configuration. The minimal configuration to operate BASIC 3 interactively is the same as the minimal configuration for INTERCOM under NOS/BE.

NOTES AND CAUTIONS

Because dynamically allocated strings cannot be implemented without changing the object code generated by BASIC, the object code generated by BASIC 3.1 is not fully compatible with that generated by later versions of BASIC. BASIC 3.1 relocatable binaries will not execute under the later version BASIC library. BASIC 3.1 programs maintained in object form must be recompiled under the later version of BASIC before they can be executed under the later versions. However, no source code changes or conversions are required.

INSTALLATION PARAMETERS

IPARAM symbol IP.CSET is used to control which character set, 63 or 64, BASIC is assembled to support.

The COMPASS micro MODLEVEL is used to specify the level of the BASIC compiler output in generated relocatable binary decks. The value of MODLEVEL is controlled by the ML parameter on the COMPASS control statement. Refer to the COMPASS Reference Manual for details.

There are four installation options controlled by BASIC symbols. Release values and Update modifications required to change them are as follows.

1. BL(burstable listing) default (release value=0; that is, listing is not burstable)
   ```
   *DELETE LIPARAM.4
   IP.BL    CEQU  1    BL DEFAULT=1 (BURSTABLE)
   ```

2. AS (ASCII parameter) default (release value=0; that is, not ASCII)
   ```
   *DELETE LIPARAM.5
   IP.AS    CEQU  1    AS DEFAULT=1(ASSUME ASCII)
   ```

3. Array base default (release value=1)
   ```
   *DELETE BASCOMP.202
   BDFLT    DATA    0    DEFAULT ARRAY ORIGIN=0
   ```

4. Messages giving time/memory required to compile/execute (release value is 1; that is, messages are on)
   ```
   *DELETE MESSAG.1
   MESSAG   EQU  0    TURNS OFF MESSAGES
   ```

NOTE

If IPARAM symbols are defined for burstable listing and ASCII options, they will override the conditional EQUs for BASIC.

INSTALLATION PROCEDURES

BASIC is conditionally assembled for NOS/BE. The IPARAM symbol OS.NAME must be SCOPE in order to generate the NOS/BE variant of the compiler.

You may obtain installation decks PL57I, PL57E, and PL57V from the Installation Decks PL using the procedure described in part I, section 1.

Job PL57I is a maintenance deck which can be used to create a revised release format tape containing a modified program library and assembled binary. Job PL57E can be used to enter BASIC into the running system or user libraries through EDITLIB, either from the released tape or from the tape created by deck PL57I.

VERIFICATION PROGRAM

The verification program supplied with the release (PL57V) compiles and executes two BASIC programs in batch mode. The first verifies that the compiler has been installed correctly and the second that the relocatable version of the runtime system has been installed correctly.

Less than 1 minute is required to run the verification program deck after BASIC has been installed.

RELEASE DESCRIPTION

Data Base Utilities (DBU) run under the NOS/BE operating system in conjunction with CDCS and QU to create a log file of changes to a data base and to recover or restore the data base.

The logging part of DBU runs in the field length with a CDCS job and adds 2000 (octal) words (plus CRM buffers) to its field length. DBU logging is added to an absolute overlay in QU.

The minimum hardware configuration for recover (entry point DFRCV) and restore (entry point DFRST) is the minimum required by the operating system. Minimum execution field length is 62K (octal). Typical is 75K (octal).

RELEASE MATERIALS

DBU is released on program library tape PL58. The structure of the release tape is as follows.

File 1	Program library
File 2	Logging binary, relocatable format
File 3	Recover/restore binary, relocatable format
File 4	Recover/restore binary, absolute format

INSTALLATION PROCEDURE

You may obtain installation decks PL58I, PL58E, and PL58V from the installation decks PL using the procedure outlined in part I, section 1.

Deck PL58I serves as a program library maintenance deck in that it allows regeneration of the DBU program library and binary files. Deck PL58E uses EDITLIB to enter DBU into the running system or user libraries either from the release tape or from a tape created by deck PL58I.

LIMITATIONS

Installation of DFRCV (recover) and DFRST (restore) requires that Sort/Merge and the FORTRAN interface to Sort/Merge be installed.

COMMUNICATIONS CONTROL PROGRAM 1.1 23

SECTION 23. COMMUNICATIONS CONTROL PROGRAM 1.1 HAS BEEN REMOVED.

RELEASE MATERIALS

COBOL version 5 release material consists of a magnetic tape identified as PL60. The structure of PL60 is as follows.

File 1	Update program library of the compiler and object routines
File 2	Relocatable binary records of the compiler
File 3	Absolute binary records of the compiler texts
File 4	Relocatable binary records of the object-time routines
File 5	Termination capsule binaries
File 6	Termination dump relocatable binaries
File 7	Absolute binary records of the compiler that may be installed on the system or onto a user library
File 8	Absolute binary record of the termination capsules
File 9	Termination dump absolute binaries

LIMITATIONS

The ANSI Communications Facility is not available.

Most user programs written for COBOL 4 require translation before they will compile and execute properly under COBOL 5. A conversion aid is available. Installation of this conversion aid is described in section 25. Refer to the COBOL 4 to COBOL 5 Conversion Aid Reference Manual for a full description of this product.

The support of 63-character collating sequences is achieved by replacing the collating character % (octal value 63) with the character : (octal value 00), when the installation selects the 63-character set option. The pseudo-names CDC-64 and ASCII-64 in the ALPHABET clause then refer to the CDC-63 and ASCII-63 collating sequences, respectively.

INSTALLATION PARAMETERS

CB5TEXT selects symbol definitions from IPTEXT for use by the COBOL 5 compiler. There are no direct references to any IPTEXT symbols within the compiler or object routines, thus allowing your site greater flexibility in changing your normal installation parameters for COBOL 5.

Symbols governing machine type, character set, and CMU option are obtained from IPTEXT, while those governing CDCS, the default page size and print density, and the default error termination must be changed within the product.

To override the system defaults in these areas, make the following changes.

To generate a compiler that will generate code for a CMU machine, insert the following after the *OPTION= and before the OP.BDP label in the CB5TEXT deck.

```
OP.BDP     CEQU     OP.YES
```

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

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

To activate CDCS1 processing, change the statement with label OP.DCS in deck CB5TEXT to the following.

```
    OP.DCS     CEQU      OP.DCS1
```

Also change the DEF CB5$CDCS statement in deck ASSEMOP.

```
    DEF      CB5$CDCS       ≠CDCS1≠ ;
```

Both of the above changes must be made or the results are unpredictable. These changes are activated of DMGMNT is = DEFINEd during the extraction of installation job PL60I.

To activate CDCS2 processing, change the statement with label OP.DCS in deck CB5TEXT to the following.

```
    OP.DCS     CEQU      OP.DCS2
```

Also change the DEF CB5$CDCS statement in deck ASSEMOP.

```
    DEF      CB5$CDCS       ≠CDCS2≠ ;
```

Both of the preceding changes must be made or the results are unpredictable. These changes are activated if DMGMNT and CD2 are =DEFINEd during the extraction of installation job PL60I.

To change the CPU type that code is generated for (and object routines are assembled for); it may be set to OP.6400 for a machine with a unified CPU or to OP.6600 for a machine with a nonunified CPU; insert the following statement after the *OPTION = statement and before the OP.MODEL label in deck CB5TEXT.

```
OP.MODEL     CEQU    OP.machine
```

To change the default organization for actual key (AK), direct access (DA), or indexed (IS) files from version 2 (ORG=NEW) to version 1 (ORG=OLD), change the DEF CB5$xxOLDNEW statement in ASSEMOP to read the following. Only the routine PROCTAB need be compiled.

```
DEF     CB5$xxOLDNEW     ≡≠OLD≠≡;      (xx is AK, DA, or IS)
```

COMPILER PROGRAM LIBRARY STRUCTURE

Because the compiler was written in two languages (SYMPL and COMPASS), the order of programs on the compiler program library differs from the order in which the programs are loaded.

The PL is divided into a number of sections, by type of deck and overlay. The common decks are first, the texts second, and so on. Within each section, the decks are in alphabetic order. Common decks that call other common decks follow the common decks that are called.

End of records provided by decks named CWEORn separate the texts, COMPASS compiler code, SYMPL compiler code, the compiler skeleton, and the object routine sections.

The compiler skeleton contained in the deck -SKEL- manages the order of loading the routines. A COPYLM is performed against the assembled -SKEL- deck using all of the compiler binaries. This results in a binary file in the correct load order.

A full Update of the program library writes the texts, the COMPASS compiler code, the SYMPL compiler code, the skeleton, and the COMPASS object routine code to the compile file. The texts are used to assemble the COMPASS and SYMPL code, and CB5TEXT is used later to supply error messages (via the PP routine DOO) at object time. The SYMPL compiler code calls the appropriate common decks to obtain installation parameter definitions.

When all has been assembled, the COMPASS and SYMPL compiler code is run against -SKEL- using COPYLM to produce the following overlay structure.

0,0	COBOL5 and other control routines. Also in this overlay are CYBER Record Manager, Common Memory Manager, and the compiler table pointers.
1,0	Compiler initialization and control statement processing
2,1	Source statement scanning. This phase scans the source statements, processes COPY statements, and produces CTEXT for use in later phases of the compiler.
2,2	PICTURE analysis. Each PICTURE clause is broken up into internal information and checked for legality.
2,3	Data base translator. Processes CDCS Sub-schema information.
5,0	IDENTIFICATION DIVISION, ENVIRONMENT DIVISION, and DATA DIVISION (except for reports) lexical analysis. The CTEXT produced by 2,1 is processed into compiler internal tables.

6,0	DATA DIVISION storage analysis. Program storage is preallocated in this phase.
7,0	REPORT SECTION parsing and lexical analysis
11,0	REPORT SECTION pseudo-code generation. The GTEXT (pseudo-code) necessary to produce the specified reports is produced.
12,0 through 12,7	PROCEDURE DIVISION parsing and lexical analysis. The CTEXT for the PROCEDURE DIVISION is digested and GTEXT for each statement is produced.
14,0	Literal pooler
16,0	Cross-reference formatter
20,0	Code generation root overlay. Contains tables, pointers, and service routines common to the code generators and the assembler.
20,1	Code generation initialization, file table and data storage generation.
20,2	Code generation. GTEXT produced by earlier overlays results in OTEXT input to the assembler.
20,3	Compiler assembler. OTEXT from 20,2 is turned into CDC CYBER machine instructions and the binaries are written out.
20,4	Debugging aids. This is a null overlay unless the compiler is assembled in debug mode.
20,5	Data Map formatter and terminal dump file producer.
30,0	Diagnostic formatter

INSTALLATION INSTRUCTIONS

The compiler installation decks provide a way of placing COBOL 5 either on the regular system library or on a user library for checkout purposes.

The first job, PL60I, does a full Update and assembly of the compiler and object routines. It produces a new tape with the same structure as the release tape, and optionally, either creates or updates user libraries.

Since a full assembly of the compiler is a lengthy process (up to 2 hours clock time on a model 73), a second job, PL60I1, is provided which, using the most recent PL60 output tape, does an UPDATE,N. and assembles only the routines modified. It then executes COPYLM against the existing binaries to produce an updated compiler. This is useful if there are problems with only a few routines or if the CDCS interface requires changing. This job produces a tape and libraries the same as PL60I, provided that none of the CRM and CMM interfaces have changed and that all decks are properly ordered on the compile file.

A third job, PL60E, replaces the compiler and object routines on the running system or user libraries with those on the tape created by either PL60I or PL60I1. You can then run the verification program, PL60V, to assure correct installation.

The most efficient method for producing upgraded COBOL 5 binaries, assuming that no code modifications have been made or are necessary since the last full assembly, is to execute the PL600 job. This job produces a new tape for which the binaries can be introduced into the running system or user libraries through PL60E. Use PL600 to reformat the overlays by using relocatable binaries from the last PL60I output tape.

The use of the LOCLIB parameter in the installation decks (refer to part I, section 1) provides users with a much more flexible approach to compiler maintenance. Users might choose, for example, to use job PL60I to apply PSR code from a PSR mini-tape against their current compiler tape, creating a user library as well as a new tape. They now have a user library which can be easily tested (a memo to their users can usually supply that), and a tape that matches the library. After users are satisfied that the new compiler has no regressions, they can run job PL60E to EDITLIB the new version onto the system.

Running the same job (PL60E) with a =DEFINE LOCLIB card and using the old PL tape produces a user library for backup. When the new compiler is running trouble-free, the old compiler user library can be purged.

If the compiler is run from a file (not a user library), it produces binaries with LDSET(LIB=COB5LIB) directives in them. These may produce nonfatal LOADER diagnostics if the object routines are not present on a library with that name.

The compiler installation job decks PL60I, PL60I1, PL600, and PL60E, as well as the verification program PL60V, can be obtained from the Installation Decks PL using the procedures outlined in part I, section 1.

Assemble the following compiler routines when either the CDCS feature or the CMU option is to be turned on.

To activate CDCS, assemble

CGENTXT, DAIO, ETABLES, GBRANCH, GIO, NEXTR, SFETS, TABLES in COMPASS

All SYMPL texts

All SYMPL routines referencing the common deck ASSEMOP

All SYMPL routines in the 2,1 and 2,3 overlays

To activate CMU, assemble

CGENTXT, GANMOVE, GCMUMOV, GCONDIT, GETIPS, GMOVAN, GMOVLIT, GMOVSA, COBTIME,

GMOVSUB, NEXTR, PUTPRFX, SFETS, TABLES in COMPASS

Nothing in SYMPL

Reassemble all object routines for both options.

RELEASE MATERIALS

The COBOL 4 to 5 Conversion Aid is released on tape PL69. The structure of the release format PL69 tape is as follows.

File 1	COBOL 4 to 5 Conversion Aid source in UPDATE program library format
File 2	COBOL 4 to 5 Conversion Aid absolute binary
File 3	COBOL 4 to 5 copy utility absolute binary
File 4	COBOL 4 to 5 Conversion Aid binary syntax file

HARDWARE CONFIGURATION

The COBOL 4 to 5 Conversion Aid requires the same minimum hardware configuration as the NOS/BE, except that execution field length may exceed that available on a 49K CM computer (refer to the following discussion under Installation Parameters).

GENERAL DESCRIPTION

The COBOL 4 to 5 Conversion Aid is a language conversion system to assist in converting CYBER COBOL 4 source programs to CYBER COBOL 5 source programs. Usage instructions for use are published in the COBOL 4 to 5 Conversion Aid Reference Manual.

INSTALLATION PROCEDURE

You can obtain jobs PL69I, PL69C, and PL69V from the installation deck program library using the procedure outlined in part I, section 1 of this document. Examine and modify these jobs to accommodate permanent file ID values as well as COBOL 4 to 5 Conversion Aid installation parameters.

The installation jobs function as follows.

- PL69I updates the program library and produces a new program library tape including the three execution time binary files. If CATALOG is defined during the extraction of PL69I, the three execution files are saved as permanent files from which they may be executed.

- PL69C copies the three COBOL 4/5 Conversion Aid binaries from either the release tape or the tape written by PL69I, and catalogs them as permanent files from which they can be executed.

- PL69V is a verification job which can be used to validate proper creation of the COBOL 4 to 5 Conversion Aid permanent files. The PL69V job uses the permanent file cataloged by either PL69C or PL69I with CATALOG defined.

INSTALLATION PARAMETERS

Installation variables for the COBOL 4 to 5 Conversion Aid can be activated during the PL69I update of the program library by the following directives.

*DEFINE CBLCOPY	Causes generation of the conversion aid version capable of handling COBOL 4 source programs containing COPY (from library statements.
*DEFINE COPLST	Causes generation of a language conversion system in which COPY statements encountered in the COBOL 4 source program are retained as real COPY statements in the COBOL 5 source. If COPLST is not defined, the COBOL 4 COPY statements are retained only as comments, and the COPY source statements, having been made available by the CBLCOP process, are inserted inline in the COBOL 5 source program.
*DEFINE LTAB	Refer to the following discussion.
*DEFINE LTAB,XLTAB	Refer to the following discussion.

THE COBOL 4 to 5 Conversion Aid may overflow some tables while converting programs with large numbers of symbols or lengthy statements. Each name table entry is a variable length of 4 + (n+9) / 10 CM words for each user-defined source program name of n characters. The COBOL 4 to 5 Conversion Aid can be reinstalled, enlarging the tables, by running job PL69I (defining LTAB or LTAB,XLTAB) and cataloging the resultant files.

Name table size relates to execution FL as follows.

	Default	*DF LTAB	*DF LTAB,XLTAB
Name table length	3200D	6500D	14000D
Execution FL	71000B	102000B	122000B

If CBLCOPY is defined, add 12000B to the preceding execution field length requirements.

Default installation is with none of the preceding symbols defined.

RELEASE MATERIALS

CYBER Control Language (CCL) is released on tape PL70. The format of the PL70 tape is as follows.

File 1 CCL source code in UPDATE program library format
File 2 CCL absolute binary

HARDWARE CONFIGURATION

CCL requires the minimum NOS/BE configuration.

GENERAL DESCRIPTION

CCL allows a job to conditionally skip or repeat control statements and to process control statements obtained from a separate file or from a library. CCL consists of three absolute overlays with entry points and verb table entries for each CCL verb as follows.

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

NOTES AND CAUTIONS

If you must change a CCL verb because of a conflict with an existing library-resident program, change both the entry point name and the verb table entry in the associated deck.

The verbs IFE and IF are synonymous; define either or both.

INSTALLATION PARAMETERS

All installation parameters are defined in deck CCL on the UPDATE program library. The Maximum Value column shows the largest value the parameter may have.

Parameter	Released Value	Maximum Value	Description
IP.FPC	10	10	Maximum mumber of characters in a formal parameter
IP.SCS	40	80	Maximum number of characters in a parameter value specification

Parameter	Released Value	Maximum Value	Description
IP.LCS	10	10	Maximum number of characters in a label character string
IP.PNL	50	1023	Procedure nesting limit
IP.FP	50	500	Maximum number of formal parameters
IP.DPF	1		Specifies one of the following. 1 The default procedure file specified with IP.DPFN is defined. 0 No default procedure file name is defined.
IP.DPFN	PROCFIL		Default procedure file name used if not specified on the BEGIN statement.
IP.TAPO	1		Specifies one of the following. 1 Procedure file can reside on tape. 2 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.
IP.EXP	100	100	Number of operands and operators allowed in a CCL expression. For each unit that this parameter is decreased from 100, the execution size of CCL is reduced by two words.
IP.NPV	6		Value used in the calculation of size of the pattern value (PVT). The PVT stores the checklist entries for each parameter in the procedure headers. The following formula determines the size of the PVT in words. PVT = IP.NPV x IP.FP x 2

Parameter	Released Value	Maximum Value	Description
IP.RLD	1		Specifies one of the following. 1 Search a library randomly by using the library directory to find the requested procedure. A random search is usually faster than a sequential search. 0 Search a library sequentially to find the requested procedure.
IP.SCL	150		Specifies the maximum length in characters of lines in procedures. Any restrictions as to the length of a command remain in effect, but a comment following the command terminator may extend the length specified by IP.SCL.
IP.ATT	1		Specifies one of the following. 1 An automatic attach is done. If the procedure file is not local to the job, CCL attempts to attach a permanent file with the same name using the ID specified with IP.ID. 0 No automatic attach is done.
IP.ID	PUBLIC		Indicates the ID if automatic attach is specified (IP.ATT is set to 1).
IP.SYS	4		Defines the value of the symbolic name SYS in CCL expressions.
IP.VER	446		Defines the value of the symbolic name VER in CCL expressions.

INSTALLATION PROCEDURE

Obtain job decks PL70I and PL70E from the Installation Decks PL tape, as described in part I, section 1. PL70I is a maintenance job deck that can be used to generate a new PL70 tape containing a revised program library and absolute binary. PL70E enters CCL into the running system from either the released PL70 tape or the new PL70 tape generated by PL70I.

INSTALLATION PROCEDURE FOR MODEL 171 CONFIGURATION 27

The model 171 7152 Mass Storage/Magnetic Tape Controller installation is a two-step procedure. First, build a deadstart tape configured only for INTERCOM. Then use INTERCOM to initiate the installation decks.

REQUIRED MATERIALS

The following materials are required to install NOS/BE on the no-card-reader minimum configuration.

- CYBER Control Language (CCL) procedure file named AUTO (on the unconfigured deadstart tape).
- CCL procedure file named MUXCR.
- INTERCOM 5 and CCI binaries.
- Changes to the existing DST3 deck with code enabled by =DEFINE CYB171 and =DEFINE INT5.
- Installation deck named MUXCRE which installs the MUXCR procedure.

RESTRICTIONS

The following restrictions are imposed.

- The AUTO procedure can extract only decks required for installing INTERCOM and the 2550 controlware.
- Any model 171 installation options are defined by the AUTO procedure and cannot be modified by the site.
- The maximum number of ports allowed for the 2550 Multiplexer is 16; the first port is defined as empty.
- No site or station addresses can be specified for a port entry.
- Only MODE 4 and ASYNC ports can be specified.

REQUIRED MACROS

The following macros are required throughout the installation procedure.

AUTO MACRO

The format of the AUTO macro is as follows:

```
AUTO, P=deck, D=density, V=vsn, PW=password, INT=version.
```

deck	Name of the installation deck to be loaded. Options are PL14E, PL99E, DST3, or MUXCRE. Default requests the BCC tape and catalogs the installation decks subset.
density	Density of the tape being read. Default is HY (800 bpi).
vsn	Volume serial number of the BCC tape; applicable only to BCC tape. Default is BCCTAP.
password	Initial INTERCOM unrestricted password.
version	Version of INTERCOM. 5 INTERCOM 5 (default)

MUXCR MACRO

The format of the MUXCR macro is as follows:

```
MUXCR,NAME=symbol,TM=port,SP=speed,HW=flag,RN=count,PORTS=address,ESTO=est,CH=channel,
  EQP=eqp,CL=cl,TID=tid
```

symbol	COMPASS symbol associated with the MUX and EST macros. The first character must be alphabetic.
port	Character string appended to characters TM for port specification. Options are 3, 4, A4A, and B4A. Default is 0, which specifies an empty port.
speed	Line speed for port specification. Options are any valid line speed such as 110, 300, 600, 1200, and so on. The MUXCR macro specifies the default value.
flag	Flag to specify whether or not port is hard-wired. If nonzero, the port is hard-wired. Default is 0, which specifies a dial-up port.
count	Repeat count for the number of identical port entries to be specified. This value must not exceed PORTS minus 1. Default is 1.
address	Largest decimal port address which can be specified for the MUX macro. The first site address is always an empty port; therefore, only address minus 1 ports can be specified. Also, address must be greater than or equal to 2 and less than or equal to 16. Default is 2, which allows a subsequent call to specify one user port.
est	Equipment status table number for the EST macro. Default is 1.

channel	Hardware channel to which the multiplexer is connected for the EST macro. Default is 0.
eqp	Equipment number of the multiplexer for the EST macro. Default is 7.
cl	Cluster address for a MODE 4 terminal. Default is 0.
tid	Terminal identification for a MODE 4 hardwired terminal. Default is AA.

INSTALLATION PROCEDURES FOR INTERCOM 5

Use the following procedure to install the model 171.

1. Deadstart the 66x using the coldstart procedure for 66x tape controllers as described in the CYBER Initialization Package (CIP) User's Handbook.

2. Deadstart the 66x using the CIP tape.

3. Create an intermediate deadstart tape with INTERCOM capability as follows.

   a. Type the following at a clear control point n.

      n.X AUTO,INT=5.

      This generates the following request.

      REQUEST (OLDPL,HY,NORING,VSN=BCCTAP)

      The installation decks necessary for creating a deadstart tape are cataloged as a permanent file after the tape is assigned.

   b. Type the following at a clear control point n.

      n.X AUTO,P=PL14E.

      This causes installation deck PL14E to be processed, which EDITLIBs INTERCOM into the running system.

   c. Type the following at a clear control point n.

      n.X AUTO,P=PL99E.

      This causes installation deck PL99E to be processed, which EDITLIBs CCI routines into the running system.

d. Type the following at a clear control point n.

```
n.X AUTO, P=MUXCRE.
```

This causes installation deck MUXCRE to be processed, which EDITLIBs the MUXCR procedure file into the running system.

e. Type the following at a clear control point n after MUXCR has completed.

```
n. X MUXCR, NAME=MUX1, PORTS=6, ESTO=3, CH=1, EQP=5.
```

This creates two files with the following contents.

```
File 1    *IDENT MUXCR
          *I EST.1
          FE EST CH=1, EQP=5, ESTO=3.
          *C CMR

File 2    MUX1 MUX2550 6, EST=3
               EMPTY
```

f. Define ports by typing the following example at clear control points n.

NOTES

The NAME=symbol keyword cannot be specified during this step. If it is specified, the file is overwritten, and step e must be performed again.

```
n.X    MUXCR,TM=A4A,HW=1,ID=AB,CL=1
n.X    MUXCR,TM=B4A.
n.X    MUXCR,RN=2.
n.X    MUXCR,TM=3.
```

If the RN keyword is not specified, RN=1 is assumed; therefore, the total repeat count for the preceding calls is 5. This is one less than the 6 specified by the PORTS= address keyword from step e.

File 2, created in step e, has the following contents.

```
MUX1  MUX2550   6, EST=3
      EMPTY
      MODE 4    CL=1,MODE=4A,LT=HW,CODE=ASCII,ID=AB
      MODE 4    CL=0,MODE=4A,CODE=BCD
      EMPTY
      EMPTY
      ASYNC     LS=AUTO
```

g. Type the following at a clear control point n.

    n. X MUXTAB.

   This creates the terminal definition file INIDxxx from file 2.

h. Type the following at a clear control point n.

    n. X AUTO,DST3,PE

   This causes the installation deck DST3 to be processed, which assembles the new central memory resident, captures the running system, and writes a new deadstart tape at PE density (1600 cpi).

4. Deadstart (warmstart) using the new deadstart tape.

5. Type the following at a clear control point n prior to bringing up INTERCOM.

    n.X AUTO,PW=INSTALL.

   This creates a password file which allows the user to log in as follows.

    LOGIN,userid,INSTALL

   More than one user can log in, but each user must specify a unique user identifier (userid).

6. Install NOS/BE using INTERCOM.

RELEASE DESCRIPTION

INTERCOM 5 in conjunction with the NOS/BE operating system provides TTY and CRT terminals with time-shared access to CYBER 180, CYBER 170, CYBER 70, and 6000 Computer Systems. Also, remote batch jobs can be submitted from terminals equipped with a remote card reader and printer, or from a low- or medium-speed batch terminal. Programs written in the FORTRAN, COBOL, COMPASS, or BASIC languages can be submitted from a remote terminal for execution at control points; the user at the remote terminal can interact with the executing program. Program output can be routed to the line printer and card punch at the central site or to a terminal equipped with line printers or card punches. Through the system permanent file feature, input from a central site magnetic tape or card reader is available to the remote user.

HARDWARE CONFIGURATION

In addition to the minimum hardware required by the NOS/BE system, INTERCOM 5 requires the following equipment for communication and operation.

- A CRT terminal, model 214-11, 214-12, 217-11, 217-12, 217-13, 217-14, 711-10, 731-12, 732-12, 734 Remote Batch Terminal, 714-10, 714-20, or CDC CYBER 18 or a model 33, 35, or 38 KSR or ASR Teletype terminal, or a 713 TTY-compatible terminal, or a 751 TTY-compatible terminal, or a HASP workstation, IBM 2780, or IBM 3780 batch terminal.

- A dedicated 255x Network Processing Unit (NPU) on a dedicated channel for TTY and/or CRT terminals.

- Data Sets for communication between the remote terminal and central site. Teletype terminals require 103, 113, or 212 series Data Sets; CRT terminals require 201, 208, or 209 series Data Sets, or CDC 358 Transceivers. Refer to the Control Data Communications Handbook for specific details of the exact modem strapping options required by INTERCOM 5.

REQUIRED HARDWARE OPTIONS

711-10	
Data control	711-102
714-10 or 714-20	
Display (8 x 80 or 16 x 80)	714-122 or 714-123
731-12	
732-12	
Memory increment (8K bytes)	730-100
Display (16 x 80)	730-101

HARDWARE OPTIONS

Teletype	
Paper tape reader/punch	
217-11, 217-12, 217-13, 217-14	
Card reader	224-11, 12, 13, or 14
Line Printer	222-11, 12, 13, or 14
711-10	
Memory option (16 x 80 screen)	711-100
Character printer	711-120 or 711-21
714-10	
Display (8 x 80 or 16 x 80) (up to 8 additional)	714-122 or 714-123
Character printer (Up to 3)	711-120 or 711-121

RELEASE MATERIALS

INTERCOM version 5 release material consists of a magnetic tape (PL14) containing the INTERCOM program library as file one.

NOTES AND CAUTIONS

Some mode 4 CRT terminals work properly in all respects except that they do not properly process the sequence bit; generally, they send a zero sequence bit in all transmissions to the 2550 NPU. This problem is a terminal malfunction, a loose wire, or bad hardware card. Such terminals may undergo endless retransmissions of one of the first two WRITES to the CRT screen. Should this retransmission be observed regularly when a particular terminal connects, hardware support personnel should check the sequence bit (bit 24) in the station address word.

LINE SKIPPING ON 714 NONIMPACT PRINTER

When column 80 is reached, an INTERCOM-generated line skip occurs. 714 nonimpact printers have a photo cell switch set at column 80 to skip a line. Thus, when input lines exceed 79 characters, output is double-spaced. If this is not desirable, request the site customer engineer to move the photo cell switch beyond column 80.

CHARACTER SET SUPPORT

The IP.CSET display code character set selection affects INTERCOM only with respect to 63- or 64-character set selection. The BCD and ASCII printer character set default and 026/029 keypunch code default selections affect only the remote batch terminals. How it affects each one is described in each terminal reference manual.

In addition to the IP.CSET display code character set selected, INTERCOM allows users to select extended ASCII 128- or 256-character sets for communication with a mode 3 type terminal. These sets are described in the INTERCOM Reference Manual.

LIMITATIONS AND SYSTEM CONSIDERATIONS

When the CONNECT command (or CONNEC call) is used, the specified data is routed to or from the terminal each time the file is read or written. When simultaneous operations are to be performed, connect no more than one file to a terminal for interactive operations at any time.

GENERAL PROCEDURES

Installation of a complete INTERCOM system requires establishing installation parameters and installing from the INTERCOM OLDPL. Run the card deck described later at the central site to install INTERCOM. Install FORTRAN Extended 4 and COMPASS before INTERCOM is installed. Use the TDFGEN utility to create the MUX-subtable and terminal ids before bringing up INTERCOM.

When a 255x front-end is configured, place the highest-speed terminals on the lowest port and place any empty ports at the high-number port positions. Therefore, the configure 255x in the following sequence: 9600-baud terminals first; then 4800-baud terminals; followed by 2400- and 2000-baud terminals; next TTY; and, finally, empty ports. On the 255x, the lowest- numbered ports should be in the leftmost CLA positions, with higher-numbered ports to the right.

INSTALLATION PARAMETERS

Consider the following items when configuring the INTERCOM system for a particular installation.

- Parameters in the INTERCOM common deck INTCOM can be changed to affect the characteristics of INTERCOM.
- An equipment status table (EST) entry must be established for 255x to be used by INTERCOM.
- Certain tables within 1CI, 1QP, and 3TT can be set to control use of selected commands.
- Parameters in the EDITOR common decks IPFTN and IPCOM can be changed to affect the characteristics of EDITOR.
- Parameters in the multiuser job common decks MUJCOM and CMUJCOM can be changed to affect the characteristics of multiuser jobs (particularly EDITOR).

INTERCOM COMMON DECK SETTINGS

Release values are shown in the following list of INTERCOM parameters for the common deck INTCOM present on PL14. If these parameters are to be changed, place the cards containing the proper code with the CEQU macro after an *INSERT INTCOM.43 directive and insert them into the first update record of the deck PL14I. Alternate tested values are shown in parentheses.

A cross-reference listing showing the routines that reference each INTCOM symbol appears in part III of this document.

IP.FTNTS CEQU 0

This parameter specifies the installation default FORTRAN Extended 4 compiler for EDITOR. A value of zero specifies OPT=0; a value of 1 specifies Time-Sharing.

IP.IACES CEQU 11

An 11-bit field contains the user table access field and user permission bits. This value must be the same as the value for IP.IACES (refer to NOS/BE IPARAMS in this section). The entire 11-bit field is used to determine if a user has access to a specific utility or routine. The setting of IP.IACES determines how many bits, right-justified, are to be used as the access level. The remaining bits (11-IP.IACES) are used as permission bits.

User access level is an octal integer (range 0 to (2**IP.IACES)-1) and is contained in the user table after the user logs in. User's access level must be greater than or equal to the access level of the command in order to use a command.

Permission bits form a mask constant (range 0 to (2**(11-IP.IACES))-1. Each bit which is set in the command permission-bit mask must also be set in the user's permission-bit mask in order for the user to use the command.

1AJ and LOADER check permission bits and access levels for commands found in the NUCLEUS Entry Point Name Table.

A program in a library, specifically the Entry Point Name Table entry in the NUCLEUS library, has an 11-bit permission bits/access level value. In addition, only this type of command verb has one additional bit associated with it indicating whether the entry is control-statement-callable. In the EPNT entry, bits 14-4 contain the permission bits and access level required; bit 3 contains the control-statement-callable bit (0 = not control statement-callable). 1AJ checks bit 3 for all control statements.

EDITLIB allows definition of permission bits and access levels via the SETAL directive or the AL parameter of the ADD and REPLACE directives. This value is not access level; it is a 12-bit value combining permission bits (upper 11-IP.IACES bits), access level (bits IP.IACES-1), and control- statement-callable (bit 0). The upper 11 bits of this value are the required permissions and access level found in bits 14-4 of the EPNT entry.

During a PASSWRD run, a user's permissions and access level are defined via the A=acclevl parameter. This value is an 11-bit octal number combining permission bits and access level. No control statement-callable value is associated with the user's acclevl value.

IP.IACES may be given any value between 0 and 11. If IP.IACES = 0, the entire field is permission bits. If IP.IACES = 11, the entire field is access level.

Example

1. IP.IACES = 6

2. EDITLIB run with directives
   SETAL (FILES, 201)
   SETAL (ASSETS, 407)

3. PASSWRD run with directives
   ADD U=USER1, P=PASS1, A=2
   ADD U=USER2, P=PASS2, A=302
   ADD U=USER3, P=PASS3, A=3077
   ADD U=USER4, P=PASS4, A=1515
   ADD U=USER5, P=PASS5, A=0712

4. As the result of the preceding installation, the following relationships exist.

COMMAND	PERMISSIONS ASK	ACCESS LEVEL	PERMISSIONS REQUIRED
FILES	1	0	0
ASSETS	2	3	1

USER	PERMISSIONS ASK	ACCESS LEVEL	PERMISSIONS GRANTED
USER1	0	2	NONE
USER2	3	2	0,1
USER3	30B	77B	3,4
USER4	15B	15B	0,2,3
USER5	7	12B	0,1,2

	FILES	ASSETS
USER1	P0	P1, AL
USER2	ALLOWED	AL
USER3	P0	P1
USER4	ALLOWED	P1
USER5	ALLOWED	ALLOWED

USER X COMMANDS ALLOWED

Pn Denied because user lacks permission n
AL Denied because user access level too low

IP.ID CEQU 1

If one, the INTERCOM user id is used as the default permanent file id by commands STORE, FETCH, and DISCARD. If zero, the permanent file id must be specified by the INTERCOM user.

IP.IDFL CEQU 55000B

Default extended memory field length assigned to a user's program when the user has not entered a field length (EFL).

IP.IDFLE CEQU 0

Default ECS field length (in multiples of 1000_8) allowed a user if no E parameter was specified for the user's id on the password file.

IP.IHEAD CMICRO 0,(CONTROL DATA INTERCOM 5.1)

Header output by 1IM when a remote terminal dials into the INTERCOM system.

IP.IIBMN CEQU 40

Minimum number of interactive empty buffers needed; these buffers are maintained by INTERCOM.

IP.IIBMX CEQU 70

Maximum number of interactive empty buffers needed.

IP.IM3BS CEQU 4095

Default page size in characters for mode 3 terminals. This value should be $20 \le x \le 4095$.

IP.IM3LW CEQU 72

Default line length in characters for mode 3 terminals. This value should be $10 \le x \le 136D$.

IP.IPRLS CEQU 100B

Priority loss per 100_8 PRUs used by an output file after the first 100_8 PRUs (refer to IP.MPRIT description).

IP.I4ABS CEQU 1040

Default screen size in characters for mode 4A terminals. This value should be $256D \le x \le 2040D$.

IP.I4ALW CEQU 80

Default line length in characters for mode 4A terminals. This value should be $32D \le x \le 136D$

IP.I4CBS CEQU 1280

Default screen size in characters for mode 4C terminals. This value should be $256D \le x \le 2040D$.

IP.I4CLW CEQU 80

Default line length in characters for mode 4C terminals. This value should be $32D \le x \le 136D$.

IP.I27BS CEQU 400

Default block size in characters for IBM 2780 terminals. The value should be $20 \le x \le 800$.

IP.I27LW CEQU 80

Default line length in characters for IBM 2780 terminals. The value should be $10 \le x \le 80$.

IP.I37BS CEQU 400

Default block size in characters for IBM 3780 terminals. The value should be $20 \le x \le 800$.

IP.I37LW   CEQU   80

Default line length in characters for IBM 3780 terminals. The value should be $10 \leq X \leq 80$.

IP.IHSBS   CEQU   400

Default block size in characters for HASP terminals. The value should be $20 \leq X \leq 800$.

IP.IHSLW   CEQU   80

Default line length in characters for HASP terminals. The value should be $10 \leq X \leq 80$.

IP.BSIM   CEQU   1 (0)

Default bisynchronous input mode for card reader files on HASP, IBM 2780 and IBM 3780 terminals; 1 selects 026 punch codes and 0 selects 029 punch codes.

IP.IND   CEQU   1 (2)

Maximum number of active INTERCOM drivers (of any type) allowed in the system simultaneously. It may not exceed six. If the system has RDF installed, include the RDF driver in the count of active INTERCOM drivers.

IP.ISFL   CEQU   2500B

Default swap-in field length for INTERCOM. The swap-in field length is the amount of memory requested to swap in an INTERCOM command.

IP.MALOC   CEQU   4000B

A 12-bit octal value defining the allocation style for files created by a multiuser job. Bit 11 always is set to one to indicate that a permanent file device is requested. Bits 5 through 0 indicate the allocation style. This value is placed in the File Name Table entry generated for new multiuser job files, in byte C.FALLOC.

IP.MPRIT   CEQU   4000B

Maximum priority to be assigned to an output file diverted by INTERCOM. If fl is the length of the file in PRUs, the priority assigned to a file can be expressed as IPMPRIT -(IP.IPRLS * (fl-100_8)/100_8), where / denotes an integer divide.

IP.MXCOR   CEQU   2500B

Maximum field length allowed for INTERCOM buffer usage (in multiples of 100 octal words); cannot exceed 4000_8. Selecting a minimum value for IP.MXCOR uses the following computation.

IP.MXCOR = FWA + CTS + BS + ZNDL + NPORT + NTID

FWA = First word address of INTERCOM table area
= LWA + 1 of CM resident programs

CTS = Connection table space
= (number of 2550s) * (TDFGEN connection table size)

BS = Buffer space
= (number of 2550s) * (LE.IPH DR + 64 * IP.PRUB + 1)

ZNDL = Length of OND pp overlay

NPORT = Number of port entries for this FE from TDFGEN

NTID = Number of terminal identifiers defined for batch in TDFGEN

IP.PRIX CEQU 3777B (7000B)

Nonzero indicates the priority given to input files read from remote site. If zero, priority will be taken from Job card.

IP.TSL CEQU 10B

Default time limit in seconds for execution of a user's program, if the user has not entered a time limit (ETL).

IP.BUFFE CEQU 10B

Number of PRU (physical record unit) buffers allocated for each 255x (FE) entry in the multiplexer subtable. These buffers are used by 1ND to transfer remote batch data to and from disk. The buffers are referred to as PRUBs.

IP.PRUB CEQU 2

Number of PRUs allocated to each PRUB.

IP.X780 CEQU 0

Specifies the default for automatic terminal detection for the BISYNC macro. If zero, the IBM 2780 is selected; if 1, the IBM 3780 is selected.

LE.IPHDR CEQU 9

Length of the header for a PRUB. The buffer header contains the FET information and is also used by the 1ND driver to store usage statistics.

The word length of a PRUB can be determined by using the preceding three parameters in the following formula.

    Length=IP.BUFFE*((IP.PRUB*64)+LE.IPHDR+1)

Q.ILNOFC CEQU 1130B

This value is a timer for 1ND. It is used to turn an OFF line back ON. The release value is about 10 minutes.

NOS/BE IPARAMS SETTINGS

Set these parameters at *INSERT IPARAMS.15 when NOS/BE is installed (deck PL1AI).

IP.IACES CEQU 11

Defines the number of bits in the access level, for use by 1AJ and LOADER. This value must be the same as that specified for the INTERCOM parameter IP.IACES.

IP.ILCMD CEQU 1

If set to 1, the last word in the user table will store the last command entered by each user for display on the DSD Q display. If 0, it will not be used for this purpose.

IP.IUSID   CEQU   2RBA

Defines the first user id available for assignment by the program PASSWRD. The value of this parameter is determined by the number of hardwired remote batch terminals defined in the system. The hardwired remote batch terminals use one id per terminal.

This user id is the lowest available to be assigned an interactive user. Every hardwired remote batch terminal connected to the system must have its own terminal id assigned to it.

A cross mapping of referencing routines and all symbols in IPARAMS (IPTEXT) can be found in part III.

EST ENTRY

The EST table, established when deck PL1AI is run to install NOS/BE, must contain an entry for each multiplexer to be used by INTERCOM or RDF. The channel referenced in this entry must be dedicated to the multiplexers on that channel when INTERCOM or RDF is active.

The multiplexer EST entries are defined using the EST macro (refer to part II, section 1, Equipment Configuration, for EST macro definition). The parameters are:

type   FE for 255x, RM for two-port mux.

CH=   Channel for the equipment.

EQP=   Equipment number for 255x Front End.

MOD=   OFF if off; otherwise, do not use.

A typical EST entry might appear as follows:

```
*I EST.1

FE EST CH=3,EQP=5
```

This entry notifies the multiplexer driver that a 2550 with equipment number 5 is on channel 3.

A typical RM entry might appear as follows:

```
*I EST.1

RM EST CH=15
```

This entry defines the two-port multiplexer on channel 15.

MULTIPLEXER TABLES

The INTERCOM/RDF multiplexer tables, used to configure multiplexers or front-end processors, are resident in the INTERCOM area after INTERCOM or RDF is initialized. These tables define those terminals which are connected to the 2550s or two-port mux and are generated by the TDFGEN utility, described later in this chapter.

COM2CC MACRO

The COM2CC macro defines a command which is processed by an independent routine in overlay 2CC. The macro format is as follows.

```
name COM2CC L=l, P=p, B=b, MP=mp, ADDR=ad
```

l	YES	User must be logged in to use this command.
	NO	User need not LOGIN if at a hardwired terminal.
	Default	YES.
p	YES	Command may be used while in a pause state.
	NO	Command may not be used while in a pause state.
	Default	NO.
b	YES	Command allowed only at a batch terminal.
	NO	Command allowed from any terminal type.
	Default	NO.

mp — Maximum number of parameters which can follow command verb; range 0-5. If MP is specified, even MP=0, parameters in the input line are counted. If the number of parameters exceeds mp, the line is rejected as a format error. Do not specify MP when commands contain parameters over 7 characters or for commands such as MESSAGE for which parameters are meaningless.

ad — 2CC address (routine name) where this command is processed. If the AD parameter is omitted, a routine with the same name as that of the command is assumed.

MUJ MACRO

The MUJ macro defines a multiuser job. A corresponding entry must be made in muj table of 1QP. The macro has the following format.

```
name MUJ ORD1QP=ord
```

ord — 1QP MUJ ordinal. EDITOR=1, HELLO7=2; MFINT=3; VEIAF=4; others should proceed sequentially from 5.

EXPCOM MACRO

The EXPCOM macro defines a command processed by 1NP, and controls parameter processing for the command. The macro has the following format.

```
name EXPCOM B=b, P=p, MP=mp, EXPORD=ord, PRE=pre
```

b — Same as for COM2CC, except default =YES.

p — Same as for COM2CC.

mp — Same as for COM2CC.

ord — 1NP command ordinal. An entry must be added at installation to the 1NP jump table for each new EXPCOM command.

pre — Address (name of 2CC subroutine which does preprocessing (prior to extraction and validation of equipment mnemonic) for this command.

REMOTE MACRO

The REMOTE macro defines commands which manipulate the user's queue files and execution jobs, specifically the commands DROP, KILL, DIVERT, EVICT, and PRIOR. Adding such an entry requires modifications to the 2CC routine REMOTE. Anyone contemplating this course should consult the IMS.

MUJ TABLE STRUCTURE (1QP)

Define each multiuser job as defined in the command table of 2CS in the muj table of 1QP, MUJTABL. The position of an entry in MUJTABL is defined as the 1QP muj ordinal. Entries are made with the macro MUJTBL, at *B 1QP.599.

```
MUJTBL name,fl,swpin,swpout,editor
```

name — Name of the muj.

fl — Field length of muj (actual value).

swpin — Delay, in 1CI cycles (depends on IP.TICI, released for 1/2 second), between discovery of need to swap in the muj and actual entry into the scheduling queue. This value increases response time to muj requests (when the muj is swapped out) but allows requests to accumulate; so that when the muj is in, it is more likely to process multiple users. Maximum is 4095.

swpout — Delay, in 1CI cycles, between discovery of need to swap out muj and actual swap out. A high value setting essentially dedicates the muj at a control point.

editor — 1 muj EDITOR.
0 otherwise.

The parameters swpin, swpout, and editor may be null, and default values 1, 0, and 0, respectively, are assumed.

TBL ASSEMBLY OPTIONS

Ten TBL command ordinals (14-23) are reserved for users to add routines to TBL. To add a routine with entry point xxx and command ordinal 14, change the fourteenth entry of TBL table TABLE to read the following.

```
CON xxx
```

The TBL command ordinal is an index into ICPLIB. TBL tests bits 0 and 1 of table ICPLIB (12-bit entries) to determine if checks should be made for the calling program. If bit 0 is set, the calling program is a system library program. If bit 1 is set, the calling program is at an INTERCOM control point.

Include changes to the tables in routines 2CS, 1QP, and TBL in the UPDATE record at the directive */ADD CORRECTIONS HERE in installation deck PL14I.

EDITOR INSTALLATION PARAMETERS

EDITOR uses two common decks, IPFTN (FORTRAN) and IPCOM (COMPASS), to contain installation parameters. Generally, a change to one common deck requires a corresponding change to the other. With the exception of arrays which must be dimensioned for FORTRAN in common deck IPFTN, the values of installation parameters are not defined in IPFTN. IPFTN merely allocates storage for these definitions. The definitions are DATA statements in the BLOCK DATA subprogram IPFILL.

IPCOM contains EQUs which define the installation parameters. Since many parameters are of such a nature that a change in one implies a change of another, a dependency chart is included (table II-28-2) to aid the installation. EDITOR is not available if only RMF is running.

Following is a summary of the steps to be taken to change an EDITOR installation parameter.

1. Change the DATA statement in IPFILL or EQU in IPCOM, or both, as indicated by the parameter description.

2. Consult the dependency chart (table II-28-2) for any dependent installation parameters that require change, and change them as in step 1.

3. Consult the dependency chart (table II-28-2) for dimensions of arrays in IPFTN. If they are affected, change them as indicated in table II-28-3.

Additionally, EDITOR has the installation parameter IP.FTNTS defined in common deck INTCOM (refer to INTERCOM Common Deck Settings).

Any changes which cause the size of the EDITOR to increase may require an increase in the field length defined for EDITOR in the MUJTABL for 1QP.

PAGES II-28-13 THROUGH II-28-15 HAVE BEEN DELETED

In table II-28-1, -* in the Range column indicates where a parameter has essentially no absolute upper limit. The installation determines the practical upper limit based on considerations such as EDITOR size and expected number of users.

Table II-28-1. EDITOR Installation Parameters (Sheet 1 of 4)

Parameter Name	Defined In		Description	Range	Release Value
	IPFILL	IPCOM			
NLINE	X		Default first line number for CREATE, EDIT, RESEQ	6L000001 to 6L999999	6L000100
NINCR	X		Default line number increment for ADD, CREATE, EDIT, RESEQ	1-999998	10
NUAS	X	X	Number of user area buffers	1-* Large number decreases response time if there are many users	3
NBBS	X	X	Number of big buffers (used for EDIT, SAVE, RUN)	1-* Increase if many EDITs, SAVEs, RUNs anticipated	2
NPBS	X	X	Number of pool buffers. Each is 64*NPRUS words	2-* Increase when heavy file modifications or long text lines expected, generally NPBS≥NUAS	3
NUSERS	X	X	Maximum number of users simultaneously using EDITOR	1-* Vary with expected usage of EDITOR	30
NPRUS	X	X	Number of 64-word PRUs in one block in edit file If NPRUS is larger than 12D, EDITOR will not use the CMU hardware, regardless of the setting of NOS/BE 1 installation parameter IP.CMU.	1-* Large number decreases response time for commands which process large files, but it also increases amount of central memory required for EDITOR by 64 words for each pool buffer and 64 words for each user area buffer	2
NSUA	X	X	Size of user area; must be modified in IPFILL if NPRUS is changed. NSUA=69+64*NPRUS. Size Size does not include portion of user area used for tabs, return jump links, and edit file index	133-*	197

Table II-28-1. EDITOR Installation Parameters (Sheet 2 of 4)

Parameter Name	Defined In: IPFILL	Defined In: IPCOM	Description	Range	Release Value
NUASIZE	X	X	Size of user area including areas for tabs, return jump links and edit file index.	133-*	230
NPRUBUF	X		Number of words in one edit file block. Must be 64*NPRUS	64-*	128
JTABS	X		Number of word in user area which holds tab values; must be modified in IPFILL if NPRUS is changed. JTABS=69+64*NPRUS	131-*	197
JNDXHDR	X		Number of index header word in user area; must be modified in IPFILL if NPRUS is changed. JNDXHDR= JTABS+ (NTBSMAX+4)/5	132-*	199
JINDEX	X		Number of first word in edit file index in user area; must be modified in IPFILL if NPRUS is changed. JINDEX= JNDXHDR+1	133-*	200
JRJLNKS	X		Number of first word in return jump link area in user area; must be modified in NPRUS is changed. JRJLNKS= JINDEX+NSINDEX	153-*	220
NSINDEX	X	X	Number of index entries for each user's edit file	1-* Increase for editing very large files	20
NTBSMAX	X	X	Maximum number of tab settings permitted by FORMAT command	1-509 Must be ≥ NTBSFTN, NTBSCOM, NTBSCOB, NTBSALG, NTBSDEF	10
XNPCENT	X		Percent to which each block of user's edit file is filled by EDIT (Padding factor)	.01-1.00 Decrease if heavy file modification is expected	.90

Table II-28-1. EDITOR Installation Parameters (Sheet 3 of 4)

Parameter Name	Defined In		Description	Range	Release Value
	IPFILL	IPCOM			
NTABFTN	X		FORTRAN tab character	1LA-1L;	1L;
NTABCOM	X		COMPASS tab character	1LA-1L;	1L;
NTABCOB	X		COBOL tab character	1LA-1L;	1L;
NTABALG	X		ALGOL tab character	1LA-1L;	1L$
NTABDEF	X		Default tab character	1LA-1L;	1L;
NTBSFTN	X		Number of FORTRAN tabs defined	0-509	5
NEDFETS	X	X	Number of FETs used to attach an user's editfile.	1-*	10
NTBSCOM	X		Number of COMPASS tabs defined	0-509	3
NTBSCOB	X		Number of COBOL tabs defined	0-509	5
NTBSALG	X		Number of ALGOL tabs defined	0-509	5
NTBSDEF	X		Number of default tabs defined	0-509	5
NCHFTN	X		Maximum number of characters in FORTRAN line	1-510	72
NCHCOM	X		Maximum number of characters in COMPASS line	1-510	72
NCHCOB	X		Maximum number of characters in COBOL line	1-510	72
NCHALG	X		Maximum number of characters in ALGOL line	1-510	72
NCHDEF	X		Maximum number of characters in default format	1-510	72
NCHBAS	X		Maximum number of characters in BASIC line	150	

Table II-28-1. EDITOR Installation Parameters (Sheet 4 of 4)

Parameter Name	Defined In: IPFILL	Defined In: IPCOM	Description	Range	Release Value
FTNTABS	X		Consecutive stream of bits, each 12 define a tab position for FORTRAN format. Must be ascending order	1-511 (each tab)	00070012001500200023B
COMTABS	X		Same as above, for COMPASS	1-511	00130022004400000000B
COBTABS	X		Same as above, for COBOL	1-511	00100014002000240030B
ALGTABS	X		Same as above, for ALGOL	1-511	00070012001500200023B
DEFTABS	X		Same as above, for default format	1-511	00070012001500200023B
NSBB		X	Size of big buffers used for EDIT, SAVE, RUN (does not include FET)	64-* Increase for very large files	157
NDEBUG		X	Flag controls debugging mode. (Refer also to multi-user installation parameter MDEBUG)	0 = off 1 = on	0

Table II-28-2. EDITOR Dependencies

Changed Parameter	Check Parameters in IPFILL and/or IPCOM	Check Arrays in IPFTN
NLINE		
NINCR		
NUAS		MMUJTBL,MUAS
NBBS		MBBS,MBBMA
NPBS		MPBS,MPBMA,MMUJTBL
NUSERS†		MMUJTBL
NSINDEX	JRJLNKS	MUAS
NTBSMAX	JNDXHDR,JINDEX,JRJLNKS	MUAS
XNPCENT		
NTABFTN		
NTABCOM		
NTABCOB		
NTABALG		
NTABDEF		
NTBSFTN	NTBSMAX,FTNTABS	FTNTABS
NTBSCOM	NTBSMAX,COMTABS	COMTABS
NTBSCOB	NTBSMAX,COBTABS	COBTABS
NTBSALG	NTBSMAX,ALGTABS	ALGTABS
NTBSDEF	NTBSMAX,DEFTABS	DEFTABS
NCHFTN		
NCHCOM		
NCHCOB		
NCHALG		
NCHDEF		
NCHBAS		
FTNTABS	NTBSFTN	FTNABS
COMTABS	NTBSCOM	COMTABS
COBTABS	NTBSCOB	COBTABS
ALGTABS	NTBSALG	ALGTABS
DEFTABS	NTBSDEF	DEFTABS
NSBB		MBBS
NDEBUG		MDEBUG††
NPRUS	JTABS,JNDXHDR,JINDEX, JRJLNKS,NSUA	MPBS,MUAS

†When NUSERS is increased, the user should also consider changing the size of the TERMIN and TERMOUT tables in the muj subroutine MUJSUBS. Refer to the INTERCOM 4 Multiuser Job Capability Programming System Bulletin under the heading Changing Size of TERMIN and TERMOUT.

††Refer to Multiuser Job Installation Parameters.

Table II-28-3. EDITOR Array Dimensions in IPFTN

Array Name	Usage	Array Dimension
FTNTABS	FORTRAN tabs	(NTBSFTN+4)/5
COMTABS	COMPASS tabs	(NTBSCOM+4)/5
COBTABS	COBOL tabs	(NTBSCOB+4)/5
ALGTABS	ALGOL tabs	(NTBSALG+4)/5
DEFTABS	Default tabs	(NTBSDEF+4)/5
MMUJTBL	Storage needed by muj subroutine tables	4*NUSER + NBBS + NEDFETS + NUAS + 6
MUAS	User area buffers	NUAS*(size of full user area) where:
	May never exceed 4095 decimal	(size of full user (area) (NTBSMAX+4)/5 + NSINDEX +1 + NSUA + NSRJLNK
		Note: NSRJLNK should not have to be changed by an installation
MBBS	Big buffers	NBBS*NSBB + NBBS*6
MPBS	Pool buffers	NPBS*64*NPRUS
MBBMA	Big buffer management area	NBBS
MPBMA	Pool buffer management area	NPBS

EDITOR DEBUG CODE

If EDITOR encounters hardware and/or software problems, a diagnostic printout is produced. If the problem is considered fatal, all EDITOR users are detached. The content of the diagnostic printout depends on the error encountered and the setting of NDEBUG. In any event, the diagnostic printout should accompany any PSR relating to a MUJ SYSTEM ERROR.

MULTIUSER JOB INSTALLATION PARAMETERS

Multiuser jobs are not available if only RDF is running. The multiuser job (muj) subroutines use two common decks, MUJCOM and CMUJCOM. Both contain storage allocation for an array, ECSBUF. The MUJCOM deck in FORTRAN code contains a DIMENSION statement; the CMUJCOM deck in COMPASS code contains a BSS statement. This array is used by the muj peripheral processor routines, FAD, to read information from extended memory. Array length must be (n*64+1) central memory words. You can select the value of n, depending on the expected use of extended memory for storage of user swap files (if extended memory is used, n should be at least 2) and on the number of local files allowed for an INTERCOM user. As a guide, increase n by one for each 20 local files allowed per user. The upper limit for n is dependent on the amount of storage used for the extended memory buffer in the muj, and the size of the swap buffer in FAD.

The peripheral processor routine FAD contains two parameters relevant to allocation of space for ECSBUF. ECSBFLN (near FAD.659) is a COMPASS EQU instruction. It must be equated to the number of central memory words in the ECSBUF array. SWAPBF (near FAD.650) is a table FAD uses to read the ECSBUF array into PP memory. The value of ECSBFLN, and thus the size of the ECSBUF array in MUJCOM and CMUJCOM, must not be greater than 1+ (length of SWAPBF)/5.

Symbol MDEBUG in common deck CMUJCOM controls muj debugging code (0=off, 1=on). Set it to 1 if the EDITOR installation parameter NDEBUG is set to 1.

In the routine MUJFILL, the two constants NACOUNT and THRSHLD control the accounting of muj time. The value of NACOUNT determines how frequently the accounting information for a muj is obtained from the system and distributed to users attached to the muj. NACOUNT is the maximum number of user switches performed on any given user before accouting is done. Accounting is always performed on user exit from the muj. NACOUNT must be set greater than or equal to 1 and defaults to 50 decimal. The value of THRSHLD determines the minimum number of CP seconds accumulated before accounting is posted to the user. As THRSHLD is set to smaller values, accounting is more accurate, overhead is increased, and THRSHLD has more meaning. THRSHLD defaults to 5 decimal.

Installation deck PL14I also compiles the relocatable multiuser job subroutines (deckname MUJSUBS). Deck PL14E does not add them to the running system for reasons of size and expected infrequency of use. However, always include MUJSUBS on the COMPILE file when EDITOR is compiled and loaded, so that references to the muj subroutines from EDITOR are satisfied. If an UPDATE,Q is done and the EDITOR is to be modified, the UPDATE input must include a *COMPILE MUJSUBS. (EDITOR does not use FTNMUJ or COBOMUJ, the decknames for the FORTRAN Extended and COBOL muj preprocessors.)

INSTALLATION PROCEDURES

You can obtain installation job decks PL14I and PL14E from the Installation Decks program library, using the procedure outlined in part I, section 1 of this document.

Deck PL14I assembles the released program library adding the created binary to the PL tape as supplemental files. The release tape does not contain assembled binary. Deck PL14E uses EDITLIB to enter the binary created by deck INTCM1 into the running system.

Deck PL14O, applicable only in the user library method of installation, allows regeneration and replacement of absolute overlays in the user library plus creation of a new PL14 tape.

Deck PL14E suggests CM residency for selected PP routines. Sites having extended memory may wish to move some of these PP routines to extended memory by employing the method discussed in the System Extended Memory Resident Routine and Library portion of part II, section 1. Once PL14E has been run, run job DST3 to capture a deadstart tape containing INTERCOM. Decks PL14E and DST3 need not be run if the user library installation process is being followed.

No INTERCOM PP programs are required to be CM resident. However, to increase product performance, installation jobs PL14E and DST1 contain EDITLIB MOVE directives to force into CM residence some driver overlays, 3TT and its overlays, 1CI and one of its overlays, and 1QP. This group of routines and overlays requires 5200 octal words. Additionally, 2ND, 3ND, 4ND, 5ND, 6ND, 7ND, and 8ND should be CM resident to prevent INTERCOM restarts in situations (usually caused by error conditions) where the disk or ECS access to read the INTERCOM driver overlay is not available for periods of 3 seconds.

INTERCOM in an idle state uses 6400 octal words for multiplexer tables, PRUB buffers, and minimum empty buffer chains.

NPU PROGRAMS

The INTERCOM 2550 Front End NPU initializer uses the NPU multiplexer subtables to determine which variants of the NPU programs to load before the NPU driver is brought up. INTERCOM assumes the proper variants are available on the system library and are disk-resident. The general format of NPU load module names is ODy, where y is a value computed from the following values assigned to the line protocols.

1	Mode 3
2	Mode 4
4	IBM 2780/3780
8	HASP

Add the values for each different type of line configured. Convert the total (1 to 15) to hexadecimal (1 to F).

Example:

A 2550 configuration includes only mode 3 and mode 4 lines. The load module name consists of OD3, since the value of mode 3 and mode 4 lines is 3.

Note that if the proper variant is not present in the system, INTERCOM will attempt to load ODF, the variant with all four terminal types.

Two additional types of load modules are used.

OD0	Micro memory module
ODZ	Bootstrap dump routine

All of these programs are available as part of a separate release library for CCI 3 that includes the NPU programs. Add NPU binaries to the running system using the following job.

```
Job statement.
EDITLIB(SYSTEM)
7/8/9
READY(SYSTEM,OLD)
REPLACE(*,INPUT)
COMPLETE.
ENDRUN.
7/8/9
Binary decks of NPU programs.
6/7/8/9
```

Section 29 of this part describes CCI 3 installation in detail.

MUJ SYSTEM ERRORS

INTERCOM multiuser jobs (for example, EDITOR), upon encountering hardware and/or software errors, produce diagnostic dumps. These dumps contain a header MUJ SYSTEM ERROR xx. This message is sent to the system dayfile and to each user currently using the muj. Error codes and their significance are described in the NOS/BE Diagnostic Handbook.

MULTIPLEXER CONFIGURATION

You must define the terminals, if any, connected to each port of each multiplexer or front-end processor to be used in the system. In addition, define terminal identifiers for each hardwired batch terminal (site or cluster). You may define multiple configurations. To define these configurations, use the TDFGEN utility.

The TDFGEN utility reads your multiplexer definition statements to create permanent files, INIDxxx, which are then used by INTERCOM and/or RDF to properly address each port and terminal.

The TDFGEN utility is called by the control statement

```
TDFGEN, parameters.
```

The parameters are all optional. If omitted, the defaults are used. Parameters may be specified in any order and are separated by commas.

FID=fid	Specifies the three-character alphanumeric suffix, xxx, to be used for the configuration file, INIDxxx. The default is the mainframe id, as specified by the CMR installation parameter, HOSTID, or as specified by the TDFILE command.
I=lfn	Specifies the file from which the multiplexer definition statements are to be read. The default is file INPUT.
L=lfn	Specifies the file to which all listable output is to be written. The default is file OUTPUT.

When TDFGEN is executed in the nondebug mode (its input file does not contain a DEBUG statement), run it as a batch job. There are no restrictions when TDFGEN is run in debug mode.

The input to TDFGEN consists of one or more multiplexer definition groups and an optional DEBUG statement. Each multiplexer group consists of a multiplexer definition statement followed by its port definition statements. The optional DEBUG statement, when specified, may occur anywhere in the input stream.

The syntax of input statements conforms to COMPASS with the following restrictions:

- Each statement keyword, such as MUX2550, MUXRM, ASYNC, MODE4, is considered to be the opcode.
- Each parameter string following a keyword is considered to be the variable field and must begin before column 30 of the statement.
- Each statement may be continued on the next input card, but only one continuation card is allowed for each statement.
- The DEBUG statement may not have a nonblank location field entry.
- If a location tag is omitted, the statement keyword may begin in column 1 of the statement.

MULTIPLEXER DEFINITION STATEMENT

INTERCOM 5 recognizes only a 255x communication subsystem. It is defined with the following statement.

```
MUX2550  EST=nn,nnn

    nnn=number of ports (maximum of 256).
    nn=EST octal ordinal of this 2550.
```

RDF recognizes only a two-port multiplexer that is defined with the following statement.

```
MUXRM  EST=es,nnn.

    es=EST ordinal of this RM.
    nnn=number of ports (maximum of 2).
```

The number of ports parameter indicates the highest number port+1 which INTERCOM or RDF is to service on that multiplexer.

PORT DEFINITION STATEMENT

Currently, four types of ports are recognized by the TDFGEN utility.

ASYNC	Any mode 3 terminal (TTY 33/35/37/38,713)
MODE4	Any mode 4 terminal (200UT, 711, 714, 214, 217)
BISYNC	IBM 2780, IBM 3780, or HASP terminal
EMPTY	Empty port (not serviced by INTERCOM 5)

NOTE

Port 0 must be empty for a 2550. Only ASYNC ports are supported on a two-port mux.

PORT DEFINITION PARAMETERS

The ASYNC port definition defines asynchronous terminals and has the following format.

```
ASYNC  keyword=xx,keyword=yy
```

Keywords include the following.

LT Line type, informational only for a two-port multiplexer.

DU Dial-up (default)

HW Hardwired

LS Line speed, informational only for a two-port multiplexer.

110	110 baud
150	150 baud
300	300 baud
600	600 baud
1200	1200 baud
2400	2400 baud
4800	4800 baud
9600	9600 baud
19200†	19200 baud
38400†	38400 baud
AUTO	Automatic baud rate recognition (110 to 2400 baud)

CO Carrier signal attribute, not applicable for two-port multiplexer.

ON Carrier initially on (default).

OFF Carrier initially off. This value is for hardwired modems that do not have a carrier-on signal until the terminal is connected.

LO Line ordinal. This parameter allows you to specify the line ordinal of the port being defined. Use of this parameter allows a site to omit EMPTY port definitions, since it causes generation of empty port definitions, if required. Line ordinals in subsequent macros must ascend in value, but need not be in sequence. Macros using the LO parameter can be mixed with macros omitting it.

AS Intended for CDC use. Setting AS=ON causes the TAPE,ON command to be simulated, allowing terminal I/O to be governed by X-OFF/X-ON sequences for use by terminal simulators.

The MODE4 port definition defines mode 4 synchronous terminals and has the following format.

MODE4 keyword=xx,keyword=yy,...

Keywords include the following:

MODE Mode of terminal:

4A	Mode 4A terminal (214, 217, 200UT)
4C	Mode 4C terminal (711 C/D, 714)
AUTO	4A/4C automatic terminal detection (default)

LT Line type:

DU Dial-up (default)
HW Hardwired

†Requires special product number 65370.

CARR	Carrier type (HW line type only) CONTR Controlled carrier (default) CONST Constant carrier
CODE	Character code of terminal (mode 4A only) ASCII ASCII character set (default) BCD BCD character set
CL=(s_1,s_2,..., s_n)	Cluster address A list of cluster (site) addresses indicates the port is to service a multidrop line to which terminals at those site addresses can be connected. Up to 12 site addresses can be specified in any order. Omission of cluster address causes the macro to assume cluster address 0.
TA=(t_1,t_2,..., t_n)	Terminal address (mode 4C, automatic terminal detection only) A list of terminal (station) addresses indicates the terminal has several CRT stations to be serviced. Printer stations must not be specified in the macro call. Up to nine terminal addresses, 1 through 3, 5 through 7, 9 through 11, can be specified in any order. Terminal addresses 0, 4, and 8 are reserved for printer stations. Omission of terminal address causes the macro to assume terminal address 0 on mode 4A ports and terminal address 1 on mode 4C or automatic terminal detection ports.
LO	Line ordinal. Refer to the ASYNC macro for description.
ID=(xx,xx,...,xx)	Preassigned terminal id. An id is required for each cluster defined on a hardwired line. Each id is a two-character alphanumeric string and must be less than the value defined for IP.IUSID. Required for hardwired terminals.

Examples of mode 4 terminal definitions follow.

```
MODE 4  LT=HW,CARR=CONST,MODE=4A,ID=AB
```

Defines a mode 4A, hardwired, constant carrier terminal with terminal id AB.

```
MODE4   MODE=4C, LT=HW, CL=(0, 5, 2), TA=(6, 1,2, 9), ID=(AB,AC,AD)
```

Defines a mode 4C, hardwired port with three cluster addresses, each of which may have four terminal addresses and three terminal ids: AB, AC, AD.

```
MODE4   MODE=4A, CL=(0, 1, 2, 5, 6)
```

Defines a multidrop mode 4A party line with five possible cluster (site) addresses.

The BISYNC port definition defines a bisynchronous terminal (IBM 2780, IBM 3780, HASP) and has the following format.

```
BISYNC  Keyword=xx, keyword=yy,...
```

Keywords include the following:

MODE Mode of terminal

2780	IBM 3780 terminal
3780	IBM 3780 terminal
HASP	HASP terminal
AUTO†	Default (IBM 2780, IBM 3780, or HASP terminal)

LT Line type

DU	Dial-up (default)
HW	Hardwired

CARR Carrier type (HW line type only)

CONTR	Controlled carrier (default)
CONST	Constant carrier

LO Line ordinal. Refer to the ASYNC macro for description.

ID=XX Preassigned id for this port. The id must be a two-character alphanumeric string and be less than the value assigned to IP.IUSID. Required for BISYNC terminals.

Examples of the BISYNC input statement include the following.

```
BISYNC  MODE=HASP, LT=HW, ID=AB.
```

Defines a HASP hardwired terminal with the terminal id AB.

```
BISYNC  LT=DU, MODE=2780
```

Defines an IBM 2780 dial-up terminal

†Defining AUTO ensures that both the IBM 2780 or 3780 and HASP TIPS are loaded into the 255x. To save memory in the 255x, AUTO should not be used if only HASP or only IBM 2780 or 3780 terminals are to be configured.

DEBUG STATEMENT

The DEBUG statement has the format:

```
DEBUG
```

When this statement appears in the input file, the debug mode is affected. TDFGEN will read and check all multiplexer definition and port definition statements for proper syntax and consistency, but the permanent file INIDxxx will not be created or modified. This allows the TDFGEN input to be checked without the danger of altering the actual multiplexer configuration definitions in any way.

When INTERCOM is first initiated, the INTERCOM initialization routine, 1I1, initiates the drivers as dictated by the multiplexers defined in the EST and the port definitions defined in the INIDxxx file. If all equipments (multiplexers) on a channel are turned off when INTERCOM is initiated, no driver is initiated to service that channel; however, the multiplexer subtables for all of the equipment will be examined and initialized by 1I1.

After the password file is established and the time has been initialized, bring up INTERCOM at control point zero with the console type-in INTERCOM. The INTERCOM system is then ready to service remote terminal users.

This is an example of a muxtable definition for a 2550 with 47 ports. (Note use of hexadecimal constants.)

```
MUX1    MUX2550 EST=5,0≡30
        EMPTY
        MODE4 LO=1,LT=HW,CODE=ASCII,MODE=4A,ID=AB
        BISYNC MODE=HASP
        ASYNC LT=HW,LO=0≡20
        ASYNC LO=57B
```

COMMAND TABLE STRUCTURE (1CI OVERLAY 2CS — COMMON DECK COMTBL)

Prior to INTERCOM installation, release values in the command table in 2CS can be changed or a new command or multiuser-job entry can be added. The command table is split into four parts based on the length of the command name. Insert new entries at the following locations (figure II-28-1).

1- or 2-character name	*I,RBS0033.9
3- or 4-character name	*I,RBS0033.18
5- or 6-character name	*I,COMTBL.58
7-character name	*I,IN40844C.14

The four command types each have an entry-definition macro as follows.

COM2CC	Defines a command processed by 2CC.
MUJ	Defines a multiuser job.
EXPCOM	Defines a remote-batch command processed by 1NP.
REMOTE	Defines a command which manipulates queue files or executing jobs.

A command-definition entry has the following general form.

name MACRO parameters

where name is the command name, such as ON, and MACRO is one of the preceding macro names.

PASSWORD FILE CREATION

Access to the INTERCOM system is controlled by passwords. The user must specify a valid password to log into the INTERCOM system. Two types of passwords exist: restricted and unrestricted passwords.

With restricted passwords, when logging in, the user must specify a valid username associated with the given password. You define valid username/password combinations. A user id (two alphanumeric characters) is assigned by you or the PASSWRD utility, and it is permanently associated with the username/password. This user id is assigned from a pool of available user ids; it is marked as available again only when the username/password is deleted.

With unrestricted passwords, the user may specify any username when logging in. The username is not validated. However, when a user first logs in under a given username, a user id is associated by the LOGIN utility with that username/password combination. Therefore, this user id is associated with the username/password combination, until the username/password is deleted from the system.

Through the INTERCOM routine PASSWRD, the installation defines valid restricted username/password combinations and valid unrestricted passwords and accounting values to be associated with the username/passwords or passwords. PASSWRD must be called from a data deck submitted to the central site as a batch job. The routine creates a permanent file (or edits an existing file). The file, with the permanent filename PASSWORDS, an ID=INTERCOM, contains a bit map defining assigned user ids, all unrestricted passwords, all restricted username/passwords, and all accounting information. It also contains all unrestricted username/password combinations. Installations with many users should do the following:

- Instruct users of unrestricted passwords always to use the same character string for username when logging in.

- Make use, on a regular basis, of the editing facilities in PASSWRD to delete all unrestricted usernames, and so on, freeing user ids.

While a user is logging in, he is assigned a temporary id. Temporary ids begin with a special character.

Use the following deck structure to debug the PASSWRD input without making any changes to the password file.

```
Job statement.
PASSWRD.
7/8/9
DEBUG
OLD or NEW
.
.
.
6/7/8/9
```

Use the following deck structure to modify the existing password permanent file.

```
Job statement.
PASSWRD.
7/8/9
OLD
ADD or
CHA or
DEL
.
.
.
6/7/8/9
```

This mode of PASSWRD operation updates the existing permanent file by adding new entries, changing existing entries, or deleting old entries. If the file does not exist, the run is changed to NEW mode.

To protect against unauthorized modification of the password file, the PASSWRD utility requests permission from the console operator before any modifications are made.

Between the NEW (or OLD) statement and the 6/7/8/9 statement appear the parameter statements that specify the new entries or the editing requirements. After a NEW statement, only ADD parameter statements may appear; after an OLD statement, either ADD or DEL parameter statements may appear. The ADD statement creates a new entry, or replaces an old entry with the same username/password. The CHA statement modifies an existing entry. The DEL statement deletes one or more entries. Use the NEW statement to delete existing files and construct new ones.

The DEBUG statement, if used, must be the first statement read by PASSWRD. It verifies that PASSWRD input is of the correct format without changing the password file.

The following shows the format of an ADD parameter statement.

```
ADD U=username,P=password,F=flength,T=time,A=acclevl,N=nfiles,E=ecsfl,I=id,R=rdf,
V=validate,D=disconnect,W=vename
```

username	Username (1 to 10 alphanumeric characters) must be specified for restricted passwords; it must be blank or omitted for unrestricted passwords.
password	Password (1 to 10 alphanumeric characters) must be specified. It must be the only unrestricted password of this name defined by the installation. If it is restricted, it must be the only username/password of this combination defined by the installation. (If the password or username/password have been previously defined, the ADD card functions as a replace.)
flength	Maximum field length available to the user (1 to 6 octal digits). If blank or omitted, 60000 octal CM words are assumed. This value must not exceed IP.MFL.
time	Time limit for user's session (1 to 4 octal digits, also defines the maximum ETL for individual jobs). If blank or omitted, 500 octal seconds are assumed.
acclevl	Access level/permission bits for the user (0 through 3777_8 range). This value defines which programs the user can access. If blank or omitted, an access level of 5 is assumed (dependent on IP.IACES setting in common deck INTCOM).
nfiles	Number of files this user is permitted to attach as local files at one time (1 to 2 octal digits). If blank or omitted, 24 (octal) files are allowed. This value must not exceed 76_8.
ecsfl	Maximum ECS field length available to the user in multiples of 1000B (1 to 4 octal digits). If blank or omitted, zero is assumed. This value must not exceed IP.MECS.
id	INTERCOM user id to be assigned to this restricted user (unrestricted passwords may not have preassigned user ids). The user id has two alphanumeric characters and may range from IP.IUSID to, but not including, 90. (The user id range of 90 through 99 is reserved for EXPORT HS.) If no id is specified, a default id is assigned.
rdf	Allows the user to log in on an RDF terminal. The value is YES or NO; default = NO.
validate	Validates control statements only if Remote Diagnostic Facility login is allowed (R=YES). Restricts the user on an RDF terminal to use only RDF commands when V=YES. The value is YES or NO; default = YES.
disconnect	Disconnects a logged in user's terminal (asynchronous terminals only) if inactive for 15 minutes. Value can be YES or NO. If RDF login is allowed (R=YES), the default is YES; otherwise, the default is NO.
vename	User name for NOS/BE Dual-State Interactive Facility (1 to 7 alphanumeric characters).

All parameters start after column 4 on the ADD, CHA, and DEL statements. You may specify them in any order and should separate them by delimiters (special characters).

The CHA statement modifies any of the values defined in the password file. At least the username (U) and password (P) or username (U) and id (I) must be given on the CHA statement. All other parameters are optional and change the desired fields. Parameters that are not present are not updated.

The CHA parameter statement has the following format.

```
CHA U=username,P=password,F=flength,T=time,A=acclevl,N=nfiles,E=ecsfl,I=id,R=rdf,
V=validate,D=disconnect,W=vename,S=newuser,Q=newpass
```

username	Username (1 to 10 alphanumeric characters) must be specified to update a particular user record.
password	Password (1 to 10 alphanumeric characters) must be specified (if id is omitted) to update a particular user record.
flength	Maximum field length available to the user (1 to 6 octal digits). This value must not exceed IP.MFL.
time	Time limit for user's session (1 to 4 octal digits, also defines the maximum ETL for individual jobs).
acclevl	Access level/permission bits for the user (0 through 3777_8 range). This value defines which programs the user can access.
nfiles	Number of files this user is permitted to attach as local files at one time (1 to 2 octal digits). This value must not exceed 76_8.
ecsfl	Maximum ECS field length available to the user in multiples of 1000B (1 to 4 octal digits). This value must not exceed IP.MECS.
id	INTERCOM user id for user record being updated. This must be specified if password is omitted.
rdf	Allows the user to log in on an RDF terminal. The value is YES or NO.
validate	Validates control statements only if Remote Diagnostic Facility login is allowed (R=YES). Restricts user on an RDF terminal to use only RDF commands when V=YES. The value is YES or NO.
disconnect	Disconnects a logged in user's terminal (asynchronous terminals only) if inactive for 15 minutes. Value can be YES or NO.
vename	User name for NOS/BE Dual-State Interactive Facility (1 to 7 alphanumeric characters).
newuser	New user name to replace the current user name (username). The old record is deleted and a new record is created, using the same ID as the original record.
newpass	New user password to replace the current password (password). The old record is deleted and a new record is created, using the same ID as the original record.

The DEL statement deletes one or more entries from one or both of the permanent files. It has two formats.

```
DEL U=username,P=password

DEL I=id
```

username	May take three forms: 1 to 10 alphanumeric characters, blank, or the character string *NAMES. If the first form is used, the username/password combination (restricted or unrestricted) is deleted; and the user id becomes available. If the second form is used, all entries in the two files with the given password are deleted. All user ids associated with these entries will become available; the password will no longer be defined. The third form may be used only if the specified password is unrestricted. All entries in the unrestricted password file with the given password will be deleted, and the associated ids will be made available. The password will still be defined.
password	Password to be processed. Whether an unrestricted password is deleted or not depends on the username parameter. If password is *NAMES, all usernames for all unrestricted passwords are deleted from the permanent files and the user ids for these usernames become available. The unrestricted passwords will still be defined.
id	User id; may be used as a shorthand notation to specify the username/password associated with this user id. The given username/password entry (restricted or otherwise) is deleted and the user id becomes available. If the password is unrestricted, it will still be defined.

SCED INSTALLATION PARAMETERS

When a multiuser job that uses SCED is installed, change default parameter values in SCED to reflect the requirements of the COBOL program involved. Change a value by deleting the default definition macro call and replacing it with a call to the SCED macro with the new parameter value. All macros are required.

Example:

```
*D  SCED.233     Deletes MAXUSR parameter
    MAXUSR 10    Replaces MAXUSR with new value
```

The SCED macro (parameter) calls are described in detail in the INTERCOM 5 Multiuser Job Capability Reference Manual.

Parameter	Default Value	Line to Replace
MAXUSR	30	SCED.233
USAREA	2,214	SCED.234
NUMINT	40	SCED.235
DEFBUF		No parameters No need to replace
OUTBUF	4,45 4,144	SCED.237,SCED.238

VERIFICATION PROCEDURE

INTERCOM is brought to control point zero when INTERCOM is entered at the console after the operator has entered the time.

The verification procedure cannot proceed unless a permanent file has been established containing the user passwords.

The following sample from an interactive terminal session indicates if INTERCOM is installed correctly. The underlined characters are typed by the user.

```
CONTROL DATA INTERCOM 5.1
DATE    06/27/80
TIME    09.27.22

PLEASE LOGIN
LOGIN

ENTER USER NAME- THOBBIE

               ENTER PASSWORD- MYPASSWRD

06/27/80       LOGGED IN AT 09.28.46.
               WITH USER-ID D3
               EQUIP/PORT 47/04
COMMAND- SITUATE

   THIS USER
D3-THOBBIE
   OTHER USERS

B6-HALLA       ER-IPRICE       FL-ALL155
FM-ALL156      BC-OPS          FI-TAYLOR
F3-ZEE         BA-4800BAUD     BB-4801BAUD
BD-HSBT        BE-MSBT         GU-SVLANX
GY-CHESLEY     HN-EBROTH       G4-JGM
   BATCH TERMINALS
AS-MODE 4A    AU-MODE 4C       AV-MODE 4A

AF-HASP       AG-MODE 4A
COMMAND- ASSETS
```

```
ASSETS OF D3 AT 09.30.11.
EQUIP/PORT              47/04
FILE QUOTA                 20
FILES IN USE                0
MAX FL                0077700
TIME LIMIT               7000
CP TIME                  .164
COMMAND- ETL,100

COMMAND- MAP,ON

COMMAND- ASSETS

ASSETS OF D3 AT 09.31.00.
EQUIP/PORT             47/04
FILE QUOTA                20
FILES IN USE               0
MAX FL                077700
TIME LIMIT              7000
ETL                     0100
MAP                       ON
CP TIME                  .174
COMMAND- FILES

NONE
COMMAND- LOGOUT

CPA         .198 SEC.         .198 ADJ.
SYS TIME                     1.159
CONNECT TIME      0 HRS. 5 MIN.
06/27/80 LOGGED OUT AT 09.31.49
```

RELEASE DESCRIPTION

Communications Control INTERCOM (CCI) Version 3 is the software and loadable controlware that supports the 255x Network Processing Unit (NPU) as a front end to INTERCOM Version 5 on CYBER 180, CYBER 170, CYBER 70, and 6000 Computer Systems. The CCI binary load modules reside in the NOS/BE 1 PPU library so they can be loaded into the 255x by INTERCOM.

Two release tapes are associated with CCI 3. PL99A consists of the CCI source program file (MUX firmware source and Post Link Editor initialization directives), the System Creation File (SCF), the binary load and listing files for the MUX firmware, and boot dump programs. PL99B consists of CCI 2550 binary macro load file, two intermediate files, the object file (LGO), and four build listings files.

CCI installation creates three downline load modules that resemble PPU binaries. The format of the downline load module names is ODy, where y is as follows.

0 For the micro memory load module

Z For the boot dump load module

1-F For the NPU load module, representing the sum of the assigned values for TIPs defined as follows.

ASYNC (MODE 3)	1
MODE 4	2
TIP780	4
HASPTIP	8

Table II-29-1 summarizes the module types and conditions under which each type can or must be created. Refer to the CCI Reference Manual for descriptions of the functions performed by each module type.

Table II-29-1. CCI Module Requirements

Module Type	Number Required	Installation Loaded Into	Deck(s)	Comments
Dump/bootstrap 2550	One required if 2550 is in use	All 2550's in network	PL99AI1	Required name is ODZ.
Phase 1 load 2550	One required if 2550 is in use	All 2550's in network	PL99AI1	Micromemory (MUX firmware); required name is OD0.
Phase 2 load 2550	One required if 2550 is in use	2550 for which module is configured	PL99AI2 PL99AVI	Macromemory; required name is ODx, where x can be 1 through F.

INSTALLATION OVERVIEW

The PL99AI1 installation job updates the two program libraries (PLs) on PL99A using the PSR batched corrective code and user/critical code on file OSMINIT. It also produces the phase 1 load (micromemory) and dump/bootstrap modules for the 2550. The PLs on PL99A are also used by installation jobs PL99AI2 and PL99AVI to produce all other modules.

The installation jobs use the following procedure to integrate PL99AI1, PL99AI2, and PL99AVI.

1. The job updates the SCF PL to produce a compile file. This file contains Update directive records as well as directives used by MPLIB, MPLINK, and MPEDIT later in the job when creating the CCI module(s).

2. The job updates the base PL with the Update directives on the SCF compile file to extract the decks needed to produce a particular CCI module depending upon the job being run.

    a. PL99AI2 produces decks containing input to PASCAL, MASSEM, ASSEM, MPEDIT, and MPLIB.

    b. PL99AVI only produces decks containing input to MPLIB, MPLINK, and MPEDIT.

After integrating the PLs and extracting the appropriate source decks for input to the cross processors, each installation job creates one or more CCI modules. PL99AI2 also produces an object file that is cataloged and copied to tape PL99B. From these object files, PL99AVI creates 2550 variants configured for individual NPUs. To achieve maximum flexibility when creating variants in this manner, the object file created by PL99AI2 includes all options that could be included in any variant used in the network. This mechanism is provided because the assemblies and compilations performed by PL99AI2 require significant amounts of time, whereas the object file manipulations performed by PL99AVI are relatively fast.

The Communications Control INTERCOM 3.0 (CCI for INTERCOM 5) build process produces a listing for the MPLINK/MPEDIT runs. These listings and any appropriate user dumps must accompany a submitted PSR. During any phase of a CCI build, an erroneous MPLINK dayfile message can appear, indicating that errors have been detected during execution. Examine the appropriate MPLINK listing to verify if errors are actually present.

Use the following procedure for building CCI.

1. Run PL99AI1 to produce a new PL99A tape. PL99AI3 purges extraneous files. If the ULIB approach is being used, define CCI2550 when extracting DST1; if ULIB is not defined, then each module as it is created is input into the running system via EDITLIB.

NOTE

> PL99E uses EDITLIB to insert CCI binaries into the running system. This job is necessary if ULIB is not defined when extracting the PL99 installation decks. If CCI2550 is defined, PL99E asks for one PL99A tape and one PL99B tape and then asks if there are any more PL99B tapes.

2. Enter N.GO. if there are more PL99B tapes; if not, enter N.DROP.

RELEASE MATERIALS

Release materials consist of three 7- or 9-track system standard labeled magnetic tapes. The PL99A tape contains the following files.

File Number	Record Number	File content	Name	Comments
1	1	CCI program library (including MUX firmware source and MPEDIT initialization directives)	CCI30BLD	Update PL
2	1	Build input PL	CCI30BUILD	Update PL
3	1	2550 Micro Load (MUX Firmware 1412)	MPPPUODO	PPU format downline load file ODO
4	1	2550 Boot Dump	MPPPUODZ	PPU format downline load file ODZ
5	1	2550 Firmware ZAPMP	ZAPMPOD (1-F)	Core image provided by MPEDIT
6	1	2550 Micro Load List	LIST	Print file
7	1	2550 Boot Dump List	LIST	Print file

The PL99B tape contains the following files.

File Number	Record Number	File Content	Name	Comments
1	1	2550 Macro Load	MPPPUODx	PPU format downline load file ODx, where x can be 1 through F
2	1	ZAPMP	ZAPMPODx	Core image provided by edit phase; x can be 1 through F
3	1	Symbol table	SYMTABODx	Symbol table provided by link phase; x can be 1 through F

File Number	Record Number	File Content	Name	Comments
4	1	Macro object file	LGO	Object file
5	1	Assembly list	LIST	Print file
6	1	PASCAL source listing	LIST	Print file
6	2	PASCAL object listing	LIST	Print file
7	1	MPLINK list	LIST	Print file
8	1	MPEDIT list	LIST	Print file

HARDWARE CONFIGURATION

The minimum hardware configuration to build CCI 3 requires a job field length of 77,000 octal words while running NOS/BE 1. (Running Pascal with a field length of 77K can require a long time to run on a busy system; using the 125K version is much faster.)

CCI HARDWARE REQUIREMENTS

The minimum equipment configuration required to execute CCI consists of the following.

1 2550-2 or 2551-1 Network Processing Unit which includes the following.

- 1 Multiplexer Loop Interface Adapter
- 1 Loop Multiplexer
- 1 Cyclic Encoder Board
- 1 CDC CYBER Communications Coupler

1 16K Memory Unit with 2550-2 Processor

1 Communications Line Adapter from either of the following.

- 2560-1 Synchronous CLA
- 2561-1 Asynchronous CLA

1 32K Additional Memory Unit

1 16K Additional Memory Unit

NOTES AND CAUTIONS

Assign the communications line adapter slots in the loop multiplexer in order of decreasing line transmission speeds. For example,

9600 bps line	Slot 1 (left-most slot)
9600 bpi line	Slot 2
2400 bps line	Slot 3
300 bps line	Slot 4
150 bps line	Slot 5

INSTALLATION PARAMETERS

Parameters that can be adjusted during the creation of CCI software load files are of three types.

Type 1	MPEDIT constants†
Type 2	Update DEFINE directives used during compile file creation
Type 3	SCF build input parameters†

Modify the following installation parameters in common decks PL99DEFS and PL99IN with their acceptable and default values. Numeric values preceded with a $ are given in hexadecimal.

All type 1 statements end with a semicolon; they can be followed by a comment which is preceded by a right arrow and followed by a down arrow.

TYPE 1 STATEMENTS

The syntax for items of type 1 is as follows.

    name = value;↱COMMENTS↓

There are no column restrictions.

Example:

    /C4LCBS=80;↱NUMBER OF LINES↓

†Any changes to type 1 or type 3 parameters require modifications to the installation deck oldpl.

The following are MPEDIT constants.

Card Identifier	Name	Description	Acceptable Values	Default Value
ZEXUSR.19	VM4FAIL	Number of 1/2 second intervals between polls for a failed mode terminal.	1-63	5
ZEXUSR.20	VM4CFAIL	Number of 1/2 second intervals for a failed mode multidrop terminal.	1-63	63
ZCNBTP.16	/C4LCBS	Maximum number of lines that can be configured. Central memory size must be able to accommodate number of lines specified. Must be greater than or equal to number of lines defined in INTERCOM MUX subtable.	1-254	32†(65K) 80†(81K) 80†(96K)

TYPE 2 STATEMENTS

The following Update DEFINE names are used during the Update that produces the SCF input to the CCI compile file. The DEFINEs select CCI software modules. These DEFINE directives must be specified for compile file generation of both CCI source and MPEDIT directives. This is controlled by extraction of the appropriate installation deck from the installation decks oldpl with an =DEFINE of the desired value.

†Parameter to be set at build time, depending on memory size, in deck PL99AI2 or PL99AVI for the 2550. The number of lines is set by the host at load time. These values are the default values specified in the installation decks.

Name	Description	Acceptable Value	Installation Deck Defined Values
MODE4	If defined,build in the mode 4 terminal interface program (TIP).	MODE4 or omitted	MODE4 if no other TIP defined
HASPTIP	If defined, build in the HASP TIP.	HASPTIP or omitted	Omitted
TIP780	If defined, build in the IBM 2780/3780 (BISYNC) TIP	TIP780 or omitted	Omitted
ASYNC	If defined, build in the MODE3 (TTY) TIP.	ASYNC or omitted	ASYNC if no other TIP defined
HASPSBO	If defined, a HASP batch out-put stream stopped condition is ignored. If omitted, the condition causes a disconnect after 30 seconds.	HASPSBO or omitted	Omitted
Core size	Specifies central memory size.	65K; 81K; 96K	65K if no other memory size defined
PRU size	Specifies batch size sent to host in multiples of 640 characters.	PRU1 PRU2 PRU3	PRU2 if no other PRU size defined
STATS	If defined, statistics are dispatched normally. If omitted, statistics are discarded.	STATS or omitted	Omitted
BANNUM	Specifies number of banner pages on print files.	BAN0 BAN1 BAN2	BAN2 if no other banners defined
63CSET	If defined, selects the 63-character set. If omitted, 64-character set is used.	63CSET or omitted	Omitted
TUP	Turns on on-line debugging aids.	TUP or omitted	Omitted

TYPE 3 STATEMENTS

The system variant is set up as follows.

```
=D R80050214.2 (in SCFVAR)
*ENT,LKCYC,$xxxx.
```

xxxx is any four-digit hexadecimal number.

CONFIGURATION AIDS

The following lists the 255x memory space required for available software.

Feature	Words (decimal)	Area That Can Be Paged
Basic software (required)	35 800	3 300
MODE4 (mode 4 TIP)	6 200	4 060
HASPTIP (HASP TIP)	6 100	4 000
TIP780 (IBM 2780/ 3780 BISYNC TIP)	4 850	3 300
ASYNC (TTY, mode 3 TIP)	1 500	1 240

The space left over is used for line tables, terminal tables, and dynamic buffer allocation. Guidelines for the utilization of this space are 48 words per line, 75 words per interactive device, and 750 words per batch device.

One interactive device is defined as either a CRT display/keyboard or a TTY keyboard/printer/paper tape or equivalent.

One batch device is defined as any one of the following or its equivalent.

- Line printer
- Card reader
- Card punch

The preceding figures are approximations only; exact memory utilization is a function of block size, line speed, and so on.

NOTE

Configuring a 255x for more than it can accommodate results in serious degradation of throughput. The 255x configuration is specified via both CCI 3 and INTERCOM installation parameters.

INSTALLATION PROCEDURES

Create CYBER Cross System permanent files by running either PL50I or PL50C.

2550-2 OR 2551-1 INSTALLATION (PL99A)

Execute deck PL99AI1 to do the following tasks.

- Update the CCI source program file and the system creation file.
- Create the permanent file environment for subsequent decks.
- Create the ODO and ODZ modules.
- EDITLIB ODO and ODZ into the running system if ULIB is not defined.
- Create a new PL99A output tape.

Execution of PL99AI2 can begin after the CCI source program file and system creation file have been updated (after the input tape for deck PL99AI1 is unloaded).

Run deck PL99AI2, selecting the desired combination of type 1, 2, and 3 variables described earlier. This deck creates the downline load modules ODy and writes the output tape PL99B. As soon as the MPPPULGO and PL99AI2LIST files are cataloged, subsequent variants can be built using the variant deck PL99AVI. The downline module is EDITLIBed into the running system if ULIB is not defined.

Deck PL99AVI can be run as many times as desired to create additional load module variants from the MPPPULGO file created by PL99AI2. This deck creates the downline load module ODy and writes output tape PL99B. The MPLINK and MPEDIT listings are always printed. Type 1, 2, and 3 variables apply to deck PL99AVI and should be different from variables used when running deck PL99AI2.

Deck PL99AI3 acts as a cleanup deck that purges files.

INSTALLATION DECK PL99E

Deck PL99E uses the output tapes created by other PL99 jobs to EDITLIB load modules into the running system.

CORRECTIVE CODE

Corrective code releases for CCI 3 require generation of new load modules. The method used to incorporate the CCI update depends on whether it involves changes to MPEDIT initialization directives only or to the CCI source program library as well.

If the corrective code involves changes to the CCI source program library, a complete CCI build is required. If only MPEDIT directives are affected, a new macro load module may optionally be created either by a variant build (using deck PL99AVI) or by patching. (Patching involves running the MPEDIT program using the ZAPMP (changed to ABSOLMP for MPEDIT input) and SYMTAB files as input to MPEDIT and using the new MPEDIT directives.)

Example of patching the 2550 macro load module.

```
job statement.
REQUEST(MPPPU,*PF)
REQUEST(ZAPMP,*PF)
ATTACH(ABSOLMP,zapmp-file-name,ID=id-name)
ATTACH(SYMTAB,symbol-table-file-name,ID=id-name)
MPEDIT(CSET=64)
CATALOG(MPPPU,mpppu-file-name,ID=id-name)
CATALOG(ZAPMP,zapmp-file-name,ID=id-name)
end-of-record
CONST
    /NAM$ = ODx, where x can be 1 through F;
    set constants, if any, to be used in assignment section†
BEGIN assignment section
    reset variables to be modified†
    change memory contents†
END.
end-of-file
```

VERIFICATION PROGRAMS

The verification of CCI can be divided into the verification of system generation and the verification of the on-line system.

SYSTEM GENERATION

CCI must complete the system generation procedures without error to ensure proper operation of the 255x system. Each system building phase must finish processing with no errors before the next procedure is initiated. The NOS/BE 1 programs which can detect errors during system generation include Update, MACRO Assembler, Pascal Compiler, Library Maintenance, Link Editor, and Post Edit Program. Consult the following reference manuals for the identification and explanation of specific types of errors.

Update Reference Manual

CYBER Cross System 1 Macro Assembler Reference Manual

CYBER Cross System 1 PASCAL Compiler Reference Manual

CYBER Cross System Version 1 Link Editor and Library Maintenance Programs Reference Manual

As released, CCI should complete system generation without errors. If the installation parameters are modified with care and the restrictions on them adhered to, errors should not occur while building CCI.

ON-LINE SYSTEM

Refer to the INTERCOM verification procedure.

†Refer to the CYBER Cross System Version 1 Link Editor and Library Maintenance Programs Reference Manual.

HARDWARE CONFIGURATION

CDCS 2 runs under NOS/BE and requires the same minimum hardware configuration as NOS/BE. The system control point job containing most of CDCS typically uses a field length between 100K and 134K octal words. To the user job, CDCS typically adds 1.5K octal words. AAM is not loaded at the user's control point if all AAM files are database files.

RELEASE MATERIALS

CDCS 2 is released on the program library tape PL74. The structure of the release tape is as follows:

File 1 CDCS program library, including utilities
File 2 CDCS absolute binary
File 3 DBMSTRD (master directory utility) absolute binary
File 4 DBQRFA (quick recovery file applier utility) absolute binary
File 5 DBRCN/DBRST (reconstruct and restore utilities) absolute binary
File 6 DBQRFI (quick recovery file initialization utility) absolute binary
File 7 CDCSBTF (batch test facility) absolute binary
File 8 DBREC (basic recovery utility) absolute binary
File 9 CDCS/DBU complete relocatable binary
File 10 CDCS object time routines relocatable binary

INSTALLATION REQUIREMENTS

CDCS 2 and the database utilities require BAM 1.5 and AAM 2 to be installed. CDCS 2 supports only AAM 2 files.

Installation of the master directory utility requires that DDL 3 directory access routines be installed.

To activate the interface between CDCS 2 and COBOL 5, refer to the installation procedure for COBOL 5.

Installation of the utilities DBRCN/DBRST (reconstruct and restore) requires that Sort/Merge 5 and the FORTRAN interface to Sort/Merge 5 be installed.

CDCS 2 requires DDL 3. DDL 3 is released on PL77. The installation deck PL74I expects to find the syntax table generator SYNGEN on PL77. A debug trace of CDCS activity can be obtained by using the E option on the SYMPL compilation. Flow points which trace the execution of CDCS modules from initialization to termination are generated by defining the value DEBUG when installing CDCS using PL74I. Flow points generation increases the execution size of CDCS by approximately 2500 octal words. Refer to the CDCS Internal Maintenance Specifications for details.

To activate system control point code in CMR, the CMR configuration parameter N.SBSYS must be set to a value greater than or equal to 2. This parameter defines the maximum number of subsystems; its default is zero.

INSTALLATION PROCEDURE

CDCS 2 is structured with overlay capsules to reduce execution field length requirements when certain components are not in use. In addition to a main overlay and 12 secondary overlays, 22 overlay capsules are included for accounting, attach, automatic recovery, basic recovery, constraints, database procedures, invoke, journal logging, display/operator interface, privacy, quick recovery, relations, and versions processing.

Obtain installation decks PL741, PL74E, PL740, and PL74V from the Installation Deck program library using the procedure described in part I, section 1.

Deck PL741 serves as a program library maintenance deck in that it allows regeneration of the CDCS program library and binary files. Deck PL74E uses EDITLIB to enter CDCS into the running system or user libraries, either from the release tape or from a tape created by deck PL741. PL740 uses the output tape created by PL741 to recreate the absolute binaries.

Because CDCS operates at a different control point from the user job, the EXIT and DMP instructions in Pl74V are required for maintenance and PSR submittal. In addition, the MAP,ON directive is required in the installation deck to obtain a load map to go with the dump.

VERIFICATION PROGRAMS

The CDCS 2 verification job, PL74V, builds all files and procedures necessary to execute a CDCS job. Operator actions are required at several points. Instructions are provided at these points by comments on PAUSE statements. Failure to set N.SBSYS as noted in the Installation Requirements causes failure of this job.

CDCS ACCOUNTING TABLE

A table of average central processor (CP) and input/output (I/O) seconds for different types of CDCS user requests was compiled and included in a CDCS internal table. The average values of CP and I/O seconds required by each type of request were obtained as the result of simulation runs on a CYBER 70 Model 74. Also, average values have been adjusted based on actual runs performing file creation and updating on indexed sequential files with 40 words as the record size.

When a user issues a CDCS request, such as open, read, or rewrite, the value in the appropriate table entry is retrieved and accumulated in the CDCS accounting accumulators for each individual user. Also, totals of the CP and I/O seconds used for all users combined are recorded and printed in the CDCS dayfile, at the end of the CDCS session, with the CDCS CHARGED and CDCS USED figures for the entire CDCS session. Because different environments produce different values for the average CP and I/O seconds required by each user request, CDCS provides an option for the DBA to modify these table values. The modification, if desired, can be accomplished by specifying the new values for CP and I/O as parameters on the CDCS control statement when CDCS is initialized on the system.

When CP and I/O parameters on the CDCS control statement contain time values, all the entries in the CDCS accounting table are multiplied by the ratio of the specified value over the default value for a random read on an indexed sequential file.

A second method to modify the CDCS accounting table is to change the values in the CDCS internal code and then to install CDCS with the recompiled accounting table values. This method allows users to modify any entry in the accounting table or to select certain operations for modification to suit installation needs.

List the CDCS routine DB$ACCT to see the current values in the accounting table. The format of an entry in the accounting table consists of the following four fields:

- Column 1 can contain a comma to indicate continuation of the previous line.

- The second field contains the user request code as follows:

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

- The third field in the accounting table always contains ACC as the macro identifier.
- The fourth field contains the CP and I/O times required by each user request. The parameters represent the different types of charges according to different file organization, logging, and other factors. The possible parameters are:

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

An example of an entry in the table is:

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

This means, for example, that for a random read performed on an indexed sequential file, the CP charge would be 4000 microseconds and the I/O charge would be 7000 microseconds.

RELEASE MATERIALS

DDL is released on the release tape known as PL77. The structure of the release tape is as follows:

File 1	Program library (including SYNGEN)
File 2	DDL binary; absolute format
File 3	Directory access routines; relocatable format
File 4	CDCS conversion routines; relocatable format
File 5	DDL binary; relocatable format

HARDWARE CONFIGURATION

DDL requires the same minimum hardware configuration as NOS/BE. A minimum of 50K CM is required to execute DDL.

INSTALLATION PROCEDURE

You can obtain installation job decks PL77I, PL77E, PL77O, and PL77V from the installation deck PL using the procedure outlined in part I, section 1.

Use PL77I to modify the PL, build a NEWPL, assemble and compile the entire DDL PL, and generate and save the relocatable and absolute binaries on the NEWPL. In non-ULIB mode, PL77I executes an EDITLIB to the running system DMSLIB library to make the directory access routines and conversion routines available for products that use the CDCS interface.

Use PL77E to install DDL into the running system or user libraries.

Job PL77O, applicable only in the user library method of installation, allows regeneration and replacement of absolute overlays in the user library plus creation of a new PL77 tape.

The main overlay of DDL is placed in the library NUCLEUS, the other overlays in the library SYSOVL. The directory access routines and the CDCS conversion routines are placed on the library DMSLIB.

SYNGEN, a special syntax table generator, is needed to compile DDL. SYNGEN, now on PL77, is designed to facilitate the implementation of syntax driven software. It is required to compile DDL and QU3.

You can install DDL 3.0 in the same library as DDL 2.x.

HARDWARE CONFIGURATION

CID 1 executes in the same minimum configuration required for INTERCOM under NOS/BE.

RELEASE MATERIALS

CID is released on program library tape PL82, which contains the following files.

File 1	Source code in Update program library format
File 2	Relocatable binaries
File 3	Absolute overlays

INSTALLATION PARAMETERS

The following symbols in CID can be changed to adjust table sizes. The variable n must be a positive value. No upper limit exists, but the size of CID increases with the value of n. The symbol represents a 0-8-6 punch in 026 keypunch format.

Breakpoint table; released value is 16 breakpoints.

```
*DELETE BREAKD.8
DEF BREAKTABSIZE   n ;
*DELETE BREAKZ.11
TABSIZE EQU n
```

Group table; released value is 16 groups.

```
*DELETE GROUPD.9
DEF GROUPTABSIZE   n  ;
*DELETE GROUPZ.11
TABSIZE EQU n
```

Trap table; released value is 16 traps.

```
*DELETE TRAP.9,10
DEF TRAPTABSIZE   n  ;
DEF TRAPXSIZE   m           m=n+3
*DELETE TRAPZ.11,12;
TABSIZE EQU n
XSIZE EQU m
```

The following parameter determines the size that the 54-table of program can become before requiring CID to recreate its overlays at debug time. The release value is 10_8 words of extra 54-table information.

```
*DELETE DBUGI.85
ROOM54 EQU n
```

INSTALLATION PROCEDURES

The IPARAM micro OS.NAME must be SCOPE and the Update directive *DEFINE NOSBE must be included as shown in installation job PL82I.

SECTION 33. PL/I VERSION 1 HAS BEEN REMOVED

FORTRAN COMMON LIBRARY VERSION 4 WITH POSTMORTEM DUMP UTILITY

RELEASE MATERIALS

FCL 4 is released on one reel of tape (PL8) with the following structure.

File 1	Program library of FCL 4 math and I/O routines in Update format
File 2	Relocatable binaries of FCL 4 math routines
File 3	Relocatable binaries of FCL 4 I/O routines
File 4	Program library of PMD routines in Update format
File 5	PMD postprocessor or absolute binary

INSTALLATION OPTIONS

MATHTXT and FCLTEXT select installation options from IPTEXT for use by FCL 4. No other direct references to IPARAMS exist in the product.

FCL PROGRAM LIBRARY STRUCTURE

When a full Update of the FCL PL is performed, the following records are written on the compile file.

MATHTXT	Text used to assemble math routines
Math routines	Call-by-name and call-by-value mathematical routines
FCLTEXT	Text used to assemble nonmath routines
Nonmath relocatable routines	I/O, Debug, Sort/Merge interface, and miscellaneous routines
Miscellaneous encapsulated routines	Loader directives and routines to be encapsulated

PMD PROGRAM LIBRARY STRUCTURE

When a full Update of the PMDPL is performed, one record on PMD relocatable routines is written to the compile file.

INSTALLATION DECKS

The following installation decks are supplied.

PL8I1	Performs a full Update and an assembly of the math routines.
PL8I2	Performs an assembly of the I/O routines and rewrites a release format tape with files 1 through 3 updated.
PL8E	Performs an EDITLIB of FCL 4 I/O routines from a release format tape into the running system or user libraries.
PL8EI	Performs a running system EDITLIB of FCL 4 math routines from the file cataloged by PL8I1 such that subsequent installation of SYMPL (PL6AI) is possible. This deck is not applicable to user library builds.
PL8AI	Performs a full Update and compilation of the PMD routines to generate the PMD postprocessor, which is saved with the new $PMDPL_1$ located on the FCL release tape along with the FCL PL and library routines written by PL8I2.
PL8AE	Performs an EDITLIB of the PMD postprocessor from the FCL release tape into the running system or user libraries.

RELEASE MATERIALS

FDBF is released on the release tape known as PL66. The structure of the release tape is as follows:

File 1	Program library
File 2	DDLF binary; absolute format
File 3	DML binary; absolute format
File 4	DDLF binary; relocatable format
File 5	DML binary; relocatable format
File 6	Object time routines binary

HARDWARE CONFIGURATION

FDBF requires the same minimum configuration as NOS/BE. A minimum of 60K CM is required to execute DDLF (the FORTRAN Extended 4 subschema compiler) and 45K CM to execute DML (the data manipulation language preprocessor).

NOTES AND CAUTIONS

FDBF requires installation of DDL 3. The installation deck expects to find the syntax table generator, SYNGEN, on PL77.

INSTALLATION PROCEDURE

Obtain installation job decks PL66I, PL66E, PL66O, and PL66V from the installation deck PL using the procedure outlined in part I, section 1.

PL66I does a full UPDATE, compilation, and assembly of DDLF, DML, and the object routines. It produces a new tape with the same structure as the release tape, and optionally, either creates or updates user libraries.

PL66E installs or replaces DDLF, DML, and the object routines on the running system or user libraries with those on the release tape or the tape created by PL66I or PL66O.

PL66O allows regeneration of absolute overlays. This job produces the same tape and libraries as job PL66I.

The use of the LOCLIB parameter in the installation decks causes PL66I, PL66E, and PL66O to editlib FDBF binaries into a local library instead of USERNUC/USEROV/UDMSL or the running system. This option provides users with a more flexible approach to installation.

The main overlay of DDLF resides in NUCLEUS, and the other overlays reside in SYSOVL. DML resides in NUCLEUS. The object routines reside in DMSLIB.

VERIFICATION PROGRAM

Run Job PL66V to verify the correct installation of FDBF.

SECTION 36. ALGOL 60 VERSION 5 HAS BEEN REMOVED

RELEASE MATERIALS

The FORTRAN 4/5 Conversion Aid resides on release tape PL65. The structure of the release tape is as follows.

File 1	FORTRAN 4/5 Conversion Aid source in UPDATE program library format
File 2	Conversion Aid absolute binary
File 3	Conversion Aid relocatable binary

HARDWARE REQUIREMENTS

Maintain the FORTRAN 4/5 Conversion Aid on the same hardware configuration as that required for FORTRAN Extended 4.

INSTALLATION PROCEDURES

You can obtain installation job decks PL65I, PL65E, and PL65V from the installation deck PL using the procedure outlined in part I, section 1.

PL65I uses the release tape as input to generate a new PL65 tape containing a revised program library, absolute binary, and relocatable binary file.

PL65E uses the release tape or the tape generated by PL65I to enter FORTRAN 4/5 Conversion Aid into the running system.

VERIFICATION PROGRAM

Run PL65V to verify the correct installation of the FORTRAN 4 to 5 Conversion Aid.

RELEASE MATERIALS

FORTRAN 5 is released on one reel of tape (PL63) which contains the compiler. The installation of FCL 5 (PL64) mathematical and I/O libraries is required for FORTRAN 5 execution (refer to section 39).

The structure of PL63 is as follows.

File 1	Program library of the FORTRAN 5 compiler
File 2	Relocatable binary
File 3	Relocatable binary
File 4	Absolute overlay binary

LIMITATIONS

Because all code generated by the compiler assumes the existence of the Integer Multiply hardware option, install all applicable Integer Multiply FCOs.

If FORTRAN 5 is installed on a CYBER 70 model 71, 72, or 73, or a CYBER 170 model 171, 172, 173, 174, 720, or 730 with the MODEL installation parameter (in IPARAMS) correspondingly set, the object code produced will execute properly but will not be optimal for a model 74, 175, 740, 750, 760, 865, 875, or 990. If MODEL is set to 74 or 175, the object code produced will execute properly on a model 71, 72, 73, 74, 171, 172, 173, 174, 175, 720, 730, 740, 750, 760, or 180-class mainframe but will be optimal only for the model selected. If MODEL is set to 176 and the source programs contain LEVEL 2 (direct access extended memory) statements, then the compiled object code will execute properly on models 176, 865, 875, or 180-class mainframe. For other models, source programs containing LEVEL 2 statements will not execute correctly and source programs without LEVEL 2 statements will execute correctly although not optimally.

Most user programs written in FORTRAN Extended 4 require translation before they compile properly under FORTRAN 5. Refer to section 37 for installation instructions and to the FORTRAN 4/5 Conversion Aid Reference Manual for a product description.

INSTALLATION PARAMETERS

You can obtain installation parameters by assembling FTN5TXT and/or FTN, depending on the parameters of interest. FTN contains the installation parameters for default control statement settings, control statement error processing, default file names, input/output buffer length, and compiler overlay library names. The remaining parameters are in OPTIONS (called by FTN5TXT).

Reinstall the compiler and CCG whenever parameters in OPTIONS are changed. Revise installation parameters in COMFCIP (called by decks FTN and INIT00) through a standard maintenance run (installation deck PL63I) if both FTN and INIT00 are reassembled.

INSTALLATION PROCEDURE

You can obtain compiler installation job decks PL63I and PL63E, and verification program PL63V from the installation deck program library using the procedure outlined in part I, section 1. PL63I updates the program library, producing a new program library tape including supplemental binary files. Deck PL63I updates the program library, producing a new program library tape including supplemental binary files. Run deck PL63E following PL63I but before attempting installation of the object library when the running system modification approach to building systems is used.

Deck PL63I references IPTEXT and CPUTEXT; part III of this document contains a cross reference map of referencing routines versus IPARAMS symbols. Deck PL63I also requires access to the COMPASS program library to acquire the common deck COMPCOM and the common common decks, and to the CCG program library to acquire the common code generator.

Decks PL63E and DST3 need not be run if the user library approach is being followed.

FORTRAN COMMON LIBRARY VERSION 5 WITH POSTMORTEM DUMP UTILITY

RELEASE MATERIALS

FCL 5 is released on one reel of tape (PL64) with the following structure:

File 1	Program library of FCL 5 math and I/O routines in Update format.
File 2	Relocatable binaries of FCL 5 routines.
File 3	PMD program library in Update format.
File 4	PMD postprocessor absolute.

INSTALLATION OPTIONS

MATHTXT and FCLTEXT select installation options from IPTEXT for use by FCL 5. No other direct references to IPARAMS exist in the product.

FCL5 PROGRAM LIBRARY STRUCTURE

When a full Update of the FCL PL is performed, the following records are written to the compile file:

MATHTXT	Text used to assemble math routines.
Math routines	Call-by-name and call-by-value mathematical routines.
FCLTEXT	Text used to assemble nonmath routines.
Nonmath relocatable routines	I/O, character routines, Sort/Merge interface, and miscellaneous routines.
Miscellaneous encapsulated routines	Loader directive and routines to be encapsulated.

PMD PROGRAM LIBRARY STRUCTURE

When a full Update of the PMD PL is performed, one record of PMD relocatable routines is written to the compile file.

INSTALLATION PROCEDURES

The following installation decks are supplied.

PL64I	Performs a full Update and an assembly of the FCL5 math and I/O routines to rewrite a release format tape with file 1 through 2 updated.
PL64E	Performs an EDITLIB of FCL 5 math and I/O routines from a release format tape into the running system of user libraries.
PL64AI	Performs a full Update and compilation of PMD routines to generate the PMD postprocessor, which is saved with the new PMD PL located on the FCL release tape along with the FCL5 PL and library routines written in PL64I.
PL64E	Performs an EDITLIB of the PMD postprocessor from the FCL5 release tape into the running system or user libraries.

RELEASE MATERIALS

EXPORT High Speed (HS) is released on one reel of magnetic tape (PL80) containing the EXPORT HS program library in Update format as file one.

HARDWARE CONFIGURATION

In addition to the minimum configuration required for NOS/BE, EXPORT HS requires the following:

At the central site

1 6673 or 6674 multiplexer
1 dedicated peripheral processor and channel
1 control point with 4300_8 to about 65100_8 CM words, depending upon terminal activity and hardware configuration
1 301B or 303 DATAPHONE† Data Set or CDC 358-3 transceiver

At the remote site

1 CDC 1700 remote terminal system
1 301B or 303 DATAPHONE Data Set or CDC 358-3 transceiver
1 1747 Data Set controller or 774-2 IGS console

The model of the data set or transceiver at the remote site must match that of the central site. You can add an additional 6673 or 6674 multiplexer on a second dedicated channel and peripheral processor and run at the same control point.

LIMITATIONS

When 40.8KB communication lines are used, each peripheral processor can service up to four terminals. However, when 50KB lines are used, each peripheral processor can service a maximum of only three terminals. This is a multiplexer hardware limitation.

EXPORT HS can communicate with several different IMPORT packages having similar, but not identical, commands. Informative messages or error diagnostics issued by the various IMPORT packages may differ in minor respects.

Interactive or graphics data streams are not supported. If the 1700 terminal being used has a 274/774 graphics display, any attempt to use the display or keyboard causes communications to be terminated until the IMPORT terminal is reloaded.

---

†DATAPHONE is a trademark of the American Telephone and Telegraph Company.

INSTALLATION PARAMETERS

You can change the following symbols in deck 1HS by the installation.

COPY — Used to define the number of EXPORT drivers. If COPY is set to EXP, use only one 6673/6674. If set to EXP1, two 6673/6674s are assumed and 1HS is used to drive the first. If COPY is set to EXP2, rename 1HS to 2HS and use it to drive the second 6673/6674. The release value is EXP.

CMBL — Length of the central memory I/O buffer for each data stream. These buffers are allocated only as necessary and must be at least 101_8 words long. The release value is 1001_8.

DBLEN — Default transmission buffer length in 12-bit bytes. This value must correspond to the value defined in the IMPORT system. The release value is 245_8.

EQUIPMENT STATUS TABLE

The EST, established when deck PL1AI is run to install NOS/BE, must contain an entry for each multiplexer used by EXPORT. The EST macro has the following format.

```
SC      EST          parameters  (of the form key=value)
```

Macro parameters used by EXPORT HS include the following:

CH= Channel for 6673/6674 multiplexer
EQP= Equipment number for multiplexer
MOD= OFF if off, otherwise do not use

Refer to part II, section 1, Equipment Configuration for more detail.

INSTALLATION PROCEDURES

Install EXPORT HS from PL80 with decks PL80I and PL80E.

Assemble CMR with the proper EST entries. Generate a deadstart tape with the new CMR and EXPORT HS programs.

EXPORT USE OF TERMINAL IDs

Terminal ids of 90 through 99 are reserved for EXPORT and will not be assigned by INTERCOM 5. To determine a terminal id, add the port number on the 6673/6674 DSC to 90. Therefore, port 3 will use terminal id 93. If a second 6673/6674 DSC is configured, its terminal ids will start at 94, determined in the same fashion as the first 6673/6674 DSC.

RELEASE MATERIALS

Data Catalog 2 is released on one reel of tape (PL73) with the following structure.

File 1	oldpl
File 2	Currently an empty file
File 3	Absolute binary of DCUPD
File 4	Absolute binary of DCSEL
File 5	Absolute binary of DCRPT
File 6	Absolute binary of DCGEN
File 7	Absolute binary of DCCONVT
File 8	Absolute binary of DCUTL
File 9	Absolute binary of DCIDX
File 10	Absolute binary of DCGEN
File 11	Absolute binary of DCCONGN
File 12	Relocatable binary of DCUPD
File 13	Relocatable binary of DCSEL
File 14	Relocatable binary of DCRPT
File 15	Relocatable binary of DCRET
File 16	Relocatable binary of DCCONVT
File 17	Relocatable binary of DCUTL
File 18	Relocatable binary of DCIDX
File 19	Relocatable binary of DCGEN
File 20	Relocatable binary of DCCONGN

INSTALLATION PROCEDURE

Data Catalog 2 requires installation of the COBOL 5 compiler and library.

Data Catalog 2 cannot be added to the running system. The product must run from permanent files.

You can obtain job decks PL73I and PL73C from the installation deck program library using the procedure outlined in part I, section 1.

The installation jobs function as follows.

PL73I Updates the program library with modifications to produce a new program library tape including relocatable and absolute binary files.

PL73C Calalogs the Data Catalog 2 binaries from the tape created by PL73I or the released tape as public files. The user is responsible for introducing the installation-defined password required to catalog files under ID=PUBLIC.

Because the PL73 installation jobs do not enter program binaries into libraries, the DSTn jobs are not applicable.

REMOTE HOST FACILITY (RHF)

The NOS/BE Remote Host Facility (RHF) links NOS/BE to a loosely coupled network via 380-170 network access devices (NADs) allowing transfer of permanent files and queue files between linked mainframes, access to NOS/BE magnetic tape drives, and communication between NOS/BE INTERCOM users and linked mainframes supporting the Interactive Transfer Facility.

RELEASE MATERIALS

The NOS/BE Remote Host Facility (RHF) consists of the following release materials.

PL1L	RHF subsystem program library.
PL91	RHF Applications program library.

PL1L has the following file structure.

File 1	RHF common decks program library (RHC)
File 2	RHF subsystem program library (RHF)
File 3	PP objects
File 4	NUCLEUS absolutes
File 5	SYSOVL overlays
File 6	LCNLIB relocatable binaries
File 7	RHFPL modsets source
File 8	RHCPL modsets source

PL91 has the following file structure.

File 1	Applications program library (RHP)
File 2	NUCLEUS absolutes
File 3	SYSOVL overlays
File 4	FTULIB relocatable binaries
File 5	RHPPL modsets source

INSTALLATION

The following jobs are contained on the BCC (batched corrective code) tape: PL1LI, PL1LE, PL91I, and PL91E. The job names whose last character is an I install RHF by the user library method. The job names whose last character is an E use EDITLIB to install RHF into either the user libraries or the running system. This is dependent upon ULIB being defined.

HARDWARE CONFIGURATION

RHF and its applications require the same minimum hardware configuration as NOS/BE plus a minimum of one 380-170 network access device (NAD).

Switch settings on the NAD are very important. If they are not set correctly, many problems result. Many switch settings must be correct to obtain any response from the NAD (Access code, NAD address, TCI enable, etc). 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. The CONTENTION/RESYNC parameter is set as follows:

- On any given trunk, the CONTENTION number for all NADs should be the same.
- On any given trunk, the RESYNC parameter for each NAD should be unique and less than n where n=(2 * CONTENTION NUMBER)+2. Refer to the 380-170 Network Access Device Hardware Reference Manual for further information.

NAD CONTROLWARE LOADING

The controlware for RHF (NAD) has been run through COPBC and can be EDITLIBed into a new deadstart tape. The deck name for the 380-170 NAD controlware is NAD-170. Unlike disk or tape controlware, NAD-170 is installed as a 1,1 overlay in library NUCLEUS where it can be accessed by BCLOAD.

To move a new release of the controlware from the controlware release tape to a NOS/BE permanent file, run a job of the following format:

Job	Description	
NADCW,TO,PE1.	Accounting, if necessary.	
REQUEST(TAPE1,PE)	Controlware release tape.	
REQUEST(BIN,PF)		
COPBC.		
CATALOG(BIN,NAD170CWARE,ID=CWARE)		
7/8/9		
01 77000016000000000000	Prefix Table	
01 16010446344233000000	NAD-170	
06 55555555555555555555		
01 16010446344233550317	NAD-170 CO	
01 16242217142701220555	NTROLWARE	
01 150737333446nnnnnn55	MG401-nnn	where nnn=level value
04 00000000000000000000		in display code.
01 50000101000000000000	50 Table	
7/8/9		
6/7/8/9		

When using the user library method, run the following job to install NAD controlware:

Job	Description
INADCW,TO.	Accounting, if necessary.
ATTACH(CW,NAD170CWARE,ID=CWARE)	
ATTACH(USERNUC,ID=CCT,PW=XYZ,RW=1)	
EDITLIB(SYSTEM)	
EXTEND(USERNUC)	
7/8/9	
LIBRARY(USERNUC,OLD)	
REPLACE($NAD-170$,CW,AL=0)	
FINISH.	
ENDRUN.	
6/7/8/9	

When using the running system method, run the following job to install NAD controlware:

Job	Description
INADCW,TO.	Accounting if necessary.
ATTACH(CW,NAD170CWARE,ID=CWARE)	
EDITLIB(SYSTEM)	
7/8/9	
READY(SYSTEM)	
LIBRARY(NUCLEUS,OLD)	
REPLACE($NAD-170$,CW,AL=0)	
FINISH.	
COMPLETE.	
ENDRUN.	
6/7/8/9	

NON-170 NAD CONTROLWARE INSTALLATION

Non-170 NAD controlware can be loaded in a manner similar to loading 170 NAD controlware. Examples are given for IBM, and MIN-NAD controlware.

380-370 NAD CONTROLWARE INSTALLATION (IBM)

To install 380-370 NAD controlware, use NAD-IBM for the controlware binary deckname and run the following job:

```
              Job                                Description

    NADCW,TO,PE1.                           Accounting, if necessary.
    REQUEST(TAPE1,PE)                       Controlware release tape.
    REQUEST(BIN,PF)
    COPBC.
    CATALOG(BIN,NADIBMCWARE,ID=CWARE)
    EDITLIB(SYSTEM)
    7/8/9
    01 77000016000000000000                 Prefix Table
    01 16010446110215000000                 NAD-IBM
    06 55555555555555555555
    01 16010446110215550317                 NAD-IBM CO
    01 16242217142701220555                 NTROLWARE
    01 150737334246nnnnnn55                 MG407-nnn      where nnn=level value
    04 00000000000000000000                                in display code.
    01 50000101000000000000                 50 Table
    7/8/9
    READY(SYSTEM)
    LIBRARY(NUCLEUS,OLD)
    REPLACE($NAD-IBM$,BIN,AL=0)
    FINISH.
    COMPLETE.
    ENDRUN.
    7/8/9
    6/7/8/9
```

380-110 NAD CONTROLWARE INSTALLATION (MIN)

To install 380-110 NAD controlware, use NAD-MIN for the controlware binary dockname and run the following job:

Job	Description	
NADCW,TO,PE1.	Accounting, if necessary.	
REQUEST(TAPE1,PE)	Controlware release tape.	
REQUEST(BIN,PF)		
COPBC.		
CATALOG(BIN,NADMINCWARE,ID=CWARE)		
EDITLIB(SYSTEM)		
7/8/9		
01 77000016000000000000	Prefix Table	
01 16010446151116000000	NAD-MIN	
06 55555555555555555555		
01 16010446151116550317	NAD-MIN CO	
01 16242217142701220555	NTROLWARE	
01 150737334046nnnnnn55	MG405-nnn	where nnn=level value in display code.
04 00000000000000000000		
01 50000101000000000000	50 Table	
7/8/9		
READY(SYSTEM)		
LIBRARY(NUCLEUS,OLD)		
REPLACE($NAD-MIN$,BIN,AL=0)		
FINISH.		
COMPLETE.		
ENDRUN.		
7/8/9		
6/7/8/9		

NAD CONTROLWARE INITIALIZATION PARAMETERS

Both BCLOAD and the RHF application MHF (Maintenance Host Facility) load NAD controlware. MHF loads local (380-170) NADs when activated by RHF. When called from the system console or a batch job, BCLOAD loads either remote or local NADs.

A set of controlware initialization parameters must be loaded into NAD memory along with the controlware itself (refer to the 380-170 NAD Hardware Reference Manual). These parameters are assembled into MHF and BCLOAD which append them to the NAD controlware during loading. The default values allow the use of all available NAD memory and provide for a maximum of 24 remote NADs and 35 paths, and no NAD buffer tracing.

To change any of the initialization parameters, modify the default values and reinstall BCLOAD and/or MHF. The initialization parameter defining the NAD memory size should normally be set to zero. At this setting BCLOAD and MHF will automatically determine the actual NAD memory size and reset buffer counts in order to use all available memory. This adjustment does not occur if the memory-size parameter is set to nonzero.

NAD buffer tracing can be controlled when generating the RHF configuration file. Refer to Network Configuration Statements later in this section, which describes the TRACE parameter for local NAD definitions (LNAD macro).

INSTALLATION PARAMETERS

The RHF subsystem installation parameters are described in this section. All necessary information for an installation to change a parameter is included in the description for each parameter.

ACNMAXC

The maximum number of connections that QTF can have active at any one time. Values may range from 1 to 10. Default is 4. (Lower values reduce QTF's memory requirement, but may also reduce the number of queue files transferred simultaneously.)

The current definition (acceptable to both COMPASS and SYMPL) is:

```
1         11       18    24          36
#ACNMAXC  #DEF#    4     # ACNMAXC   #4#;
```

The change deletion location is in Comdeck COMCAPR. An example of a parameter change follows:

```
*DELETE COMCAPR.xx  (This is in Comdeck COMCAPR in RHC)
#ACNMAXC  #DEF#    2     # ACNMAXC   #2#;
```

TIMEOUT

The time in seconds in which a response must be received by an application from the remote NAD/application before the connection is broken. Values may range from 1 through 1800_{10} seconds. Default is 600_{10}. The current definition format (acceptable to both COMPASS and SYMPL) is:

```
1          11      18    24         34
#TIMEOUT   #DEF#   600D  #TIMEOUT   #600#;
```

The change deletion location is in Comdeck COMCAPR. The following is an example of a parameter change:

```
*DELETE COMCAPR.xx (This is in Comdeck COMCAPR in RHC)
#TIMEOUT   #DEF#   400D  #TIMEOUT   #400#;
```

MAXRTRY

The number of retries that an application will attempt to successfully complete a file transfer is determined by this parameter. Values may range from 1 through 50_{10}. Default is 3. The current format (acceptable to both COMPASS and SYMPL) is:

```
1            11      18          24         34
#MAXRTRY     #DEF    03D         #MAXRTRY   #03#;
```

The change deletion location is in Comdeck COMCAPR. The following is an example of a parameter change:

```
*DELETE COMCAPR.xx (This is in Comdeck COMCAPR in RHC)
#MAXRTRY     #DEF#   20D         #MAXRTRY   #20#
```

MAXFILEXFR — The maximum number of file transfers that FIP allows for any one application. Values may range from 1 through 10. Default is 4. The current definition format is:

```
1     7   11          22
      DEF MAXFILEXFR  #4#;
```

The change deletion location is in Comdeck COMADEF. The following is an example of a parameter change:

```
*DELETE COMADEF.xx (This is in Comdeck COMADEF in RHF)
      DEF MAXFILEXFR  #5#;
```

FETBUFSIZE — The number of words assigned to buffer space for each file transfer. Values may be zero or larger. Default is 3200_{10}. The current definition is:

```
1     7     11          22
      DEF FETBUFSIZE  #3200#;
```

FIP will override low values.

FETBUFSIZE	ASSIGNED (binary)	ASSIGNED (coded)
0 to 969	969	1359
970 to 1359	970 to 1359	1359
1360 to	1360 to	1360 to

Values larger than 6400 (98 PRUs) do not increase transfer rates significantly and make job swapping more likely because of the increased central memory required.

The change deletion location is in Comdeck COMADEF. The following is an example of a parameter change:

```
*DELETE COMADEF.xx (This is in Comdeck COMADEF in RHF)
      DEF FETBUFSIZE  #3200#;
```

DEBUG CODE — Debug code is normally not compiled or assembled. It can be invoked by including the E and C parameters on all SYMPL compiler control directives and PC=DEBUG on all COMPASS control directives. However, any conditional debug code is not supported and is not intended for a production environment.

APPLICATION ACCOUNTING DEFINITION

RHF starts jobs that load controlware automatically, start applications in response to remote requests (QTFS, FTFS) and automatically start applications upon RHF initiation or in response to operator commands (QTF,MHF).

RHF uses the same skeleton job to build these jobs. It inserts the job name and, if appropriate, the application name and parameters. RHF routes the job to the input queue. Accounting parameters that you may need to add to the RHF skeleton job depend on the accounting method selected by your installation.

The first two statements of the default job skeleton are:

```
1         11      18
JOBPTR    DIS     ,*JOBAB,T77777.*
APLPTR    DIS     ,*application name and parameters*
```

The first statement of the change deletion location is in deck RCFGEN on RHF PL (PL1L). The following is an example of an accounting parameter change:

```
*INSERT RCFGEN.xx (where xx is the line number of the first statement of the change
                  deletion location)
          DIS     ,*ACCOUNT.    accounting parameters*
```

FTFS INSTALLATION PARAMETERS

The FTFS installation parameters have two internal constants that control the ACCOUNT statement processing.

The first constant, ACTREQ, if set to a nonzero value will force MFLINK users to specify a valid ACCOUNT control statement as the first user text directive. The default value for ACTREQ is zero. The following is an example of UPDATE directives that may be used to change the ACTREQ value:

```
1         11      18
*DELETE FTFS.xx       (where xx is the line number of ACTREQ)
ACTREQ    EQU     1
```

The second constant, ACTRFC, if set to a nonzero value will force MFLINK users to specify a valid ACCOUNT control statement before any CATALOG control statements. The default value for ACTRFC is zero. The following is an example of UPDATE directives that may be used to change the ACTRFC value:

```
1         11      18
*DELETE FTFS.xx       (where xx is the line number of ACTRFC)
ACTRFC    EQU     1
```

Before FTFS sends a job to the input queue, it replaces end-of-job control statements with the following:

```
1         11      18
          DIS     ,*APR,11.*
          DIS     ,*EXIT,S.*
          DIS     ,*PAUSE. FIFS FAILED, GO=DUMP, KILL=NO LISTING.*
```

These are defined in FTFS on RHP PL (PL91).

MAINTENANCE HOST FACILITY (MHF)

MHF starts automatically when RHF is initiated, and remains active until RHF terminates. MHF loads system-resident 380-170 microcode into local NADs, dumps and reloads local NADs that fail, and periodically copies NAD error log entries to the maintenance file (CERFILE). Parameters in the local and remote NAD definitions determine MHF's action: refer to Network Configuration Statements later in this section.

If a local NAD fails and AUTODUMP is set in its LNAD entry, MHF saves a NAD dump as the last labelled binary record of permanent file NDFxxx, ID=RHF (xxx is the mainframe identifier). Each record label contains the name NDMPLch (ch is the NAD's channel number), record number, and the date and time of the dump. MHF also puts this dump information in its dayfile:

```
MHF, NAD DUMP RECORD NAME = NDMPLch
MHF, RECORD NO.0000  TIME = hh.mm.ss
MHF, FILE = NDFxxx   DATE = mm/dd/yy
```

ITEMIZE and DMPNAD can be used to locate and list a particular dump record from the permanent file.

RHF keeps a dump count for each local NAD, to avoid unnecessarily repeated dumps. When the count exceeds 3, MHF ignores the AUTODUMP flag and proceeds with loading. The dump count can be reset by turning off the NAD's EST entry from the system console.

Deck COMMCOM on PL1L contains the definition of the first three characters of the permanent file name in the following line.

```
DEF NDFNAME     # "NDF" #; # NAD DUMP FILE NAME, CHARS 1-3 #
```

Characters 4 through 6 are set by MHF.

Deck ATTFIL on PL1L contains the definition of the permanent file identifier and passwords:

```
ID      MICRO      1,,*RHF*
CN      MICRO      1,,*RHFPERM*
RD      MICRO      1,,*RHFREAD*
TK      MICRO      1,,*RHFTURN*
```

MFLINK DELAY FILE

To maximize MFLINK performance, you must catalog a special delay file. An MFLINK job that requires local or remote RHF resources is swapped out through use of the delay file, when those resources are busy. The first MFLINK job that must wait for a resource will attach the delay file with exclusive access and remain at a control point retrying at specified intervals. Subsequently, jobs that must wait for a resource will be swapped when they attempt to attach the delay file. When the job that has the delay file attached acquires the needed resource, the delay file is released, and another job swaps in, attaching the delay file and acquiring resources.

When waiting for local RHF resources (awaiting availability of RHF or an NETON entry for MFLINK), an MFLINK job attaches and retains the delay file until the resources are available or the job is dropped. When waiting for remote RHF resources, an MFLINK job attaches the delay file and makes a specified number of attempts to connect with the remote application servicer. If after the specified number of attempts the connection is not made, the delay file is returned and the MFLINK job goes to the end of the delay queue. The delay file is now available to the next job. Upon successful connection, the delay file becomes available to the next job. In this way, jobs waiting for resources on different mainframes are given an equal opportunity to obtain them.

MFLINK jobs attach the delay file with PW=RHFPERM and RW=0. Therefore, the delay file should be cataloged with TK and XR passwords of RHFPERM to allow MFLINK to attach the delay file with exclusive access.

The ATTACH statement in MFLINK is as follows:

```
ATTACH  ZZZZZDL,DELFpid,ID=RHF,PW=RHFPERM,SN=DEFPFS,RW=0
```

The setname parameter (DEFPFS) is replaced by the default permanent file setname at execution time. This means that the delay file must reside on the default permanent file set.

MFLINK executes on the mainframe with the physical identifier pid. The permanent file name, DELFpid is used so that each mainframe in a multimainframe environment with a shared default permanent file set can have a unique delay file. The "pid" part of DELFpid is appended to the permanent file name at execution time.

An example of delay file creation for a mainframe with pid=MFA is as follows:

```
REQUEST,A,PF.
REWIND,A.
CATALOG,A,DELFMFA,ID=RHF,XR=RHFPERM,TK=RHFPERM.
```

Although MFLINK will continue to function without a cataloged delay file, the use of a cataloged delay file is strongly recommended.

RHF CONFIGURATION FILE GENERATION

You must define the RHF configuration including all NADs, applications, and LIDs/PIDs to be used by or accessible by RHF. You may define multiple configurations. Use the RCFGEN utility to define these configurations.

The RCFGEN utility reads network configuration statements to create the RHF configuration record, which RHF uses to define the network and determine proper access.

Each configuration record has a name of one to seven letters or digits, starting with a letter. The default name set by RCFGEN when creating a record (and used by RHF when searching for a record) is RCFpid,† where pid is the three-character physical identifier of the host. You can use the RH=recname parameter on the RCFGEN and RHF commands to specify a different configuration record name.

When RHF is initiated, it searches for the specified configuration record first on local file RCFILE, then on permanent file RCFpid ID=RHF, and finally on the system NUCLEUS library.

The RCFGEN utility is called by the following control statement:

```
RCFGEN,I=lfn,L=lfn,O=lfn,RN=recname,LO=OPTION.  comments.
```

The parameters are order-independent and optional. If omitted, the defaults are used.

Parameter	Description
I=lfn	Specifies the file from which the RHF configuration statements are to be read. The default is file INPUT.
L=lfn	Specifies the file to which all listable output is to be written. The default is file OUTPUT.
O=lfn	Specifies the lfn to which the configuration tables are to be written. The default is file RCFILE.
RN=recname	Specifies the name given to the configuration record being generated. The default name, RCFpid, is used if RN is omitted, if only the keyword RN is specified, or if recname is C.
LO=option	Specifies the list options used when generating listable output. If LO is omitted or option is 0, only the network configuration statements and diagnostics are listed. If only the network LO is specified, macro definitions and table-generating definitions are also listed. LO=ALL specifies all list options. (The nondefault list options may help analysis of RCFGEN or RHF problems.)
comments	Specifies an optional 1- to 70-character string which is placed in the prefix table of the configuration record being generated. (During initiation, RHF displays the string in the dayfile.)

†RCFpid is the permanent file name for the RHF configuration file, where pid is defined to be the Physical ID of the mainframe RHF is on.

The input to RCFGEN consists of network configuration statements. The syntax of input statements conforms to COMPASS macro requirements. Configuration statements (except local and remote NAD configuration statements) must not start before column 3. Use the following configuration definition statements.

Statement	Description
LNDR	Defines the maximum number of local NDRs allowed to execute at one time.
NPID	Defines the physical ID of a remote mainframe.
NLID	Defines the logical ID of a remote mainframe.
PATH	Defines the paths to a remote mainframe through the LCN Network.
RNAD	Defines the addressing information necessary to access a remote NAD.
LNAD	Defines information necessary to address local NADs.
APPL	Defines application programs that are allowed to access RHF.
DEBUG	Defines debug parameters.
CHARGE	Defines the charge that is transferred to a UCP for each RHF call.

The NLID and PATH statements must be associated with a given physical mainframe (NPID statement). The following is the required structure of these statements for each NAD statement when defining a network:

1. NPID statement.
2. All lids (NLID statement) associated with the above NPID.
3. All paths (PATH statement) associated with the above NPID.

Configuration definition statements must be specified in the following order to properly define a network.

1. LNDR statement (if necessary).
2. Application programs.
3. Repeat NPID, NLID, and PATH statements until all portions of the network are defined.
4. LNAD statements to define local NADs.
5. RNAD statements to define hardware addressing of the remote NADs.
6. DEBUG and CHARGE statements (order-independent).

NOTE

At least one of each configuration statement is required in the configuration file with the exceptions of the LNDR, DEBUG, and CHARGE statements. Defaults are specified in the individual statement descriptions.

NETWORK CONFIGURATION STATEMENTS

Following are the network configuration statements and their descriptions.

REMOTE MAINFRAME DEFINITION

Each remote mainframe definition requires 3 words in RHF's field length. To define a remote mainframe, enter:

```
NPID   PID=pid,ENABLED=status,MFTYPE=b
```

Parameter	Description
pid	Unique 3-character physical identifier of the remote mainframe (required). The characters must be alphanumeric.
status	Indicates whether the mainframe identified by PID is available. Values may be YES or NO. Default is YES.
b	1-7 character string indicating the mainframe type, e.g., NOSBE, NOS, CY200 (required). This parameter is used only to display information to the operator on the ID display.

LID DEFINITIONS FOR THE REMOTE MAINFRAME

Every two lids defined for a remote mainframe require one word in RHF's field length. To define a LID for a remote mainframe, enter:

```
NLID   LID=lid,ENABLED=status
```

Parameter	Description
lid	3-character logical identifier for the mainframe identified by the last PID definition. The LID may be the same as the PID for the last NPID configuration statement. The LID parameter is required.
status	Indicates whether the mainframe identified by PID is available using this LID. Values may be YES or NO. Default is YES.

NOTE

At least one NLID statement is required for each PID defined.

PATH DEFINITIONS TO THE REMOTE MAINFRAME

Every path defined to a remote mainframe requires two words in RHF's field length. To define a path to a remote mainframe, enter:

```
PATH   ENABLED=status,LT=tttt,RT=rrrr,RNAD=raddr,LNAD=laddr,AC=aaaa
```

Parameter	Description
status	Indicates whether the path is available when RHF is initialized. Values may be YES or NO. Default is YES.
tttt	Local trunk enables. A four-digit nonzero binary number indicating the network trunk connections for the local NAD (required).
rrrr	Remote trunk enables. A four-digit nonzero binary number indicating the network trunk connections for the remote NAD (required).
raddr	Symbolic address of the remote NAD entry for this path, referenced in the RNAD statement (required).
laddr	Symbolic address of the local NAD entry for this path referenced in the LNAD statement (required).
aaaa	A four-digit hexadecimal access code for the remote NAD. Default is 0. Access code should always be 0. If the site wants to use additional security, refer to the 380-170 Network Access Device Hardware Reference Manual. Default is 0.

REMOTE NAD DEFINITIONS

Every remote NAD defined requires four words in RHF's field length. To define a remote NAD, enter:

```
raddr    RNAD   ND=nn,DD=d,LOG=status
```

Parameter	Description
raddr	A symbolic address referenced in a preceding path statement. This parameter is required and must begin in column one or two.
nn	Two-digit hexadecimal remote device address indicating the address of the remote NAD. Default is 0.
d	One-digit hexadecimal destination device address indicating the exit port of the remote NAD. Default is 0.
status	Specifies if remote NAD trunk errors are to be recorded by MLTF in the mainframe's error log. Values may be YES or NO. Default is NO.

LOCAL NAD DEFINITIONS

Each local NAD definition requires five words in RHF's field length. To define a local NAD, enter:

```
laddr    LNAD    CH=ch,MAXNDRS=n,DEDICATE=status,CMPATHS=paths,CMBUFFS=buffers,
                 AUTODUMP=ad,AUTOLOAD=al,LOG=logstat,TRACE=trw
```

Parameter	Description
laddr	Symbolic address referenced in a preceding PATH statement. This parameter is required and must begin in column one or two.
ch	Channel (octal) that NAD is on.
n	Maximum number of NAD drivers (NDRs) that may be assigned at one time to this NAD ($1 \leq n \leq 3$). Default is 1.
status	Indicates whether the driver will always hold the NAD channel reservation between consecutive blocks of one I/O request. Values may be YES or NO. Default is YES. YES should always be specified unless some non-CDC driver requires high-performance access to the NAD channel.
paths	Maximum number of convert mode paths ($0 \leq$ paths ≤ 63). Default is 0.
buffers	Maximum number of convert mode buffers ($0 \leq$ buffers ≤ 63). Default is 0.
ad	Indicates if RHF is to automatically dump the local NAD that fails. Values may be YES or NO. Default is NO. (YES is intended for fault analysis, along with TRACE=YES/FULL, described below.)
al	Indicates if RHF is to automatically load the local NAD initially and if it fails. Values may be YES or NO. Default is YES.
logstat	Error logging status; specifies if local NAD trunk errors are to be transmitted by MHF to the mainframe's error log. Values may be YES or NO. Default is YES.
trw	Controls the value of the trace word in the NAD microcode initialization parameters used by MHF when loading the NAD. Values may be NO (no trace), YES (standard trace buffers), or FULL (maximum trace buffers). Default is NO. Refer to the 380-170 Network Access Device Reference Manual for a description of the trace word. (TRACE=YES or FULL produces trace word values of 2954 or 529F hexadecimal.) Because it decreases NAD performance, TRACE should be specified only for fault analysis. AUTODUMP=YES should be specified along with TRACE=YES/FULL to capture the NAD memory dump.

NOTE

AUTOLOAD must be YES and the NAD's EST entry ON and DOWN to allow loading of the NAD during RHF initialization.

Figure II-42-1 illustrates a sample LCN network and is the basis of the configuration statements in examples 1 and 2.

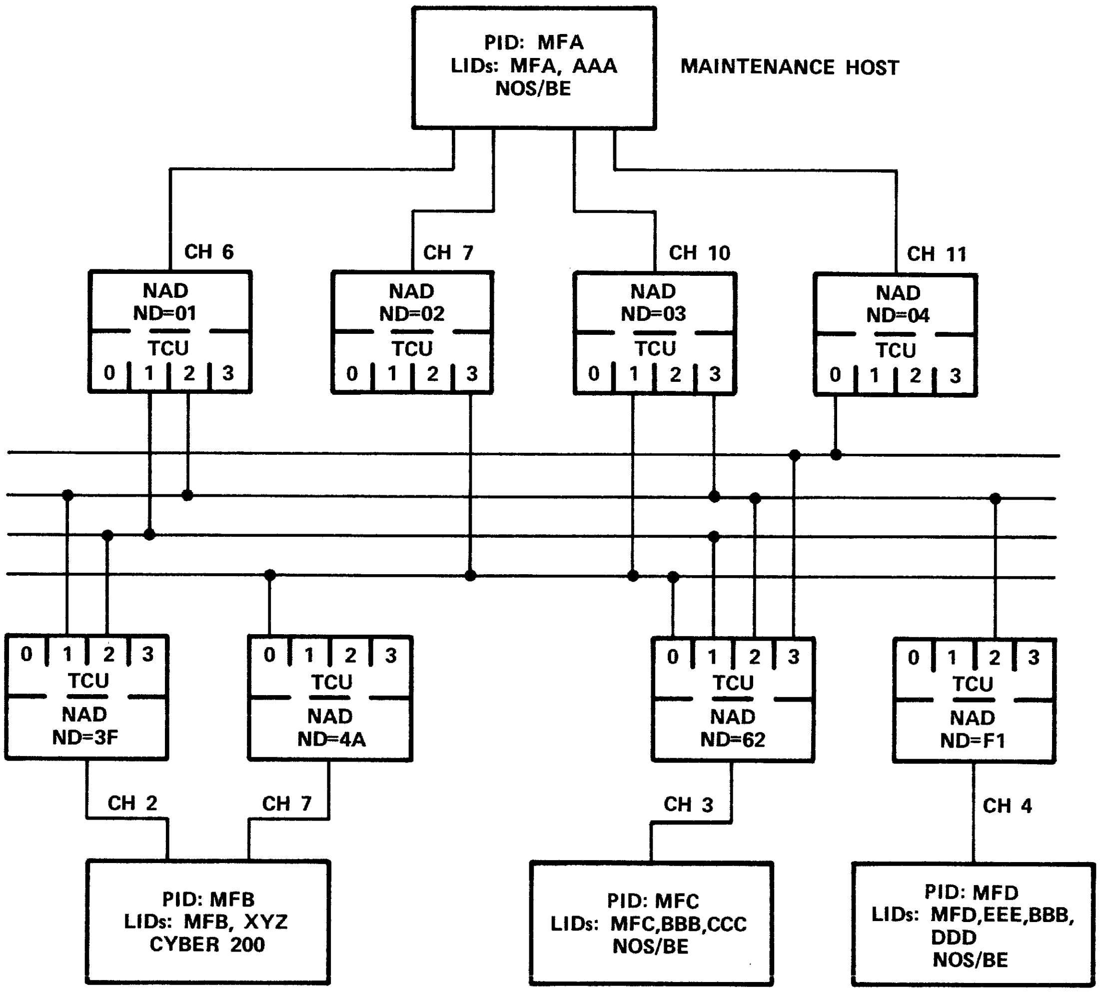


Figure II-42-1. Sample LCN Network

Example 1. The following is a sample configuration for the RHF on mainframe MFA in figure II-42-1.

```
     APPL    NAME=QTF,MXCONS=4,ASTART=YES
     APPL    NAME=QTFS,MXCONS=1,MXCOPYS=8,ASTART=NO,SVR=YES
     APPL    NAME=FTFS,MXCONS=1,MXCOPYS=8,ASTART=NO,SVR=YES
     APPL    NAME=FTF,MXCOPYS=8
     APPL    NAME=USRAP,MXCONS=6,ENABLED=NO
     APPL    NAME=ITF,MXCONS=2

     NPID    PID=MFB,ENABLED=YES,MFTYPE=C200
     NLID    LID=MFB
     NLID    LID=XYZ
     PATH    ENABLED=YES,LT=0110,RT=0110,RNAD=RN2,LNAD=LN1,AC=FOFO
     PATH    ENABLED=YES,LT=0001,RT=1000,RNAD=RN1,LNAD=LN2,AC=FOFO
     PATH    ENABLED=NO,LT=0001,RT=1000,RNAD=RN2,LNAD=LN3,AC=FOFO
     PATH    ENABLED=YES,LT=0100,RT=1000,RNAD=RN1,LNAD=LN3,AC=FOFO

     NPID    PID=MFC,MFTYPE=NOSBE
     NLID    LID=MFC
     NLID    LID=BBB,ENABLED=NO
     NLID    LID=CCC
     PATH    ENABLED=NO,LT=0110,RT=0110,RNAD=RN3,LNAD=LN1,AC=FOFO
     PATH    ENABLED=NO,LT=0001,RT=1000,RNAD=RN3,LNAD=LN2,AC=FOFO
     PATH    LT=0101,RT=1010,RNAD=RN3,LNAD=LN3,AC=FOFO
     PATH    LT=1000,RT=0001,RNAD=RN3,LNAD=LN4,AC=FOFO

     NPID    PID=MFD,MFTYPE=NOSBE
     NLID    LID=MFD
     NLID    LID=EEE
     NLID    LID=DDD
     NLID    LID=BBB
     PATH    LT=0010,RT=0010,RNAD=RN4,LNAD=LN1,AC=FOFO
     PATH    LT=0001,RT=0010,RNAD=RN4,LNAD=LN3,AC=FOFO

LN1  LNAD    CH=6
LN2  LNAD    CH=7
LN3  LNAD    CH=10
LN4  LNAD    CH=11

RN1  RNAD    DD=0,ND=4A,LOG=YES
RN2  RNAD    DD=0,ND=3F,LOG=YES
RN3  RNAD    DD=0,ND=62,LOG=NO
RN4  RNAD    DD=0,ND=F1,LOG=NO
```

Example 2. The following is a sample configuration for the RHF on mainframe MFD in figure II-42-1.

```
     LNDR      MAXNDRS=2

     APPL      NAME=QTF,ASTART=YES,MXCONS=4
     APPL      NAME=QTFS,MXCOPYS=4,SVR=YES
     APPL      NAME=FTF,MXCOPYS=6
     APPL      NAME=FTFS,MXCOPYS=6,SVR=YES
     APPL      NAME=ITF,MXCONS=2

     NPID      PID=MFA,MFTYPE=NOSBE
     NLID      LID=MFA
     NLID      LID=AAA
     PATH      RNAD=RN01,LNAD=LN1,LT=0010,RT,0010,AC=FOFO
     PATH      RNAD=RN03,LNAD=LN1,LT=0010,RT=0001,AC=FOFO

     NPID      PID=MFB,MFTYPE=NOSBE
     NLID      LID=MFB
     NLID      LID=XYZ
     PATH      RNAD=RN3F,LNAD=LN1,LT=0010,RT=0100,AC=FOFO

     NPID      PID=MFC,MFTYPE=NOSBE
     NLID      LID=MFC
     NLID      LID=BBB
     NLID      LID=CCC
     PATH      RNAD=RN62,LNAD=LN1,LT=0010,AC=FOFO

LN1  LNAD      CH=4,MAXNDRS=2

RN01 RNAD      ND=01,LOG=NO
RN03 RNAD      ND=03,LOG=NO
RN3F RNAD      ND=3F,LOG=NO
RN62 RNAD      ND=62,LOG=NO
```

LOGICAL IDENTIFIER DEFINITION AND USE

The logical identifier (lid) is the identifier used to refer to a mainframe. A user refers to a lid on an MFLINK command, a ROUTE command, or a job command.

For successful access to or from another mainframe, both the RHF configuration (including lids) and the lid table in the CMR must be set up correctly. The other (remote) mainframe likewise must have its configuration and lid tables set up properly to receive or generate a successful network access.

For either QTF or MFLINK to transfer files between mainframes, the lid used must be defined and enabled in the RHF configuration table. For either QTFS or FTFS to respond to a remote mainframe request for file transfer, the lid specified on the remote mainframe must be defined and enabled in RHF's configuration table as a valid lid for that mainframe. Likewise, that lid must be defined in the CMR lid table. For incoming connect requests, the mainframe physical identifier (HOSTID) is used as a lid.

SPECIAL LOOPBACK CAPABILITY

A special loopback capability is available on RHF for the use of both QTF and MFLINK. This capability is intended primarily for test purposes, but it may be used for other purposes as desired. The loopback capability allows a file to be sent from the local mainframe out to a NAD and back to the local mainframe instead of to a remote mainframe. To use this capability, the lids used to specify loopback must be defined properly both in RHF's configuration table and in the CMR lid table.

To allow loopback, a pid entry must be defined to match the local mainframe lid in RHF's configuration. Lids to be used for loopback should follow. Next, the loopback path should be defined. The remote NAD specified on the PATH definition should have, as an address, the address of the local NAD.

The lid to be used for MFLINK loopback, in addition to being in RHF's configuration, but must not be defined in the CMR lid table.

NOTES

The MFLINK and QTF loopback lids must be different.

Example: The following is an example of the additions to the RHF configuration file and CMR to allow loopback capabilities on mainframe MFC in figure II-42-1.

RHF configuration file:

```
        NPID     PID=MFC,MFTYPE=NOSBE
        NLID     LID=MFC
        NLID     LID=BBB
        NLID     LID=CCC
        PATH     LNAD=LN1,RNAD=RN1,LT=0001,RT=0001,AC=FOFO

LN1     LNAD     CH=3

RN1     RNAD     ND=62
```

CMR entries:

```
*INSERT,CMRIP.1
HOSTID  CMICRO,       MFC
*INSERT,LID.1
       LID            BBB
```

On mainframe MFC, lid BBB may be used for MFLINK loopback testing. On the outbound path (from MFLINK to RHF to the NAD) the NLID and PATH directives in the RHF configuration file identify which NAD the request is to be sent to. On the inbound path (from the NAD to RHF to the FTFS servicer application), the lid in the CMR ID table is used to verify that the request is valid for this mainframe. Note that since the mainframe lid MFC has also been defined as a valid lid in the RHF configuration file, you could also use MFC for MFLINK loopback.

For QTF loopback, testing, lids BBB or MFC cannot be used since their presence in the CMR ID table would cause the queue files to be processed or executed directly on the host mainframe (MFC). To allow loopback testing, QTF checks to see if the queue file destination lid is associated with the host lid in the RHF configuration file. If so, QTF will use the host lid as the transfer lid and the destination lid as the source lid.

For example, if a job is routed to the MFC input queue with a destination lid of CCC, after the file has been transferred by QTF/QTFS, it will be placed in the MFC input queue with destination lid of MFC and a source lid of CCC. Thus, by default, all output from the job will receive a destination lid of CCC and be transferred back through QTF and QTFS.

APPLICATION DEFINITIONS

Each application definition uses additional RHF field length. This additional field length is equal to MXCOPYS*(5+3*MXCONS) CM words. To define an RHF application, enter:

```
APPL   NAME=n,ENABLED=status,MXCONS=m,MXCOPYS=x,SVR=s,ASTART=a,SYSORG=sys
```

Parameter	Description
n	1-7 character name where the first character must be alphabetic and the remainder may be alphanumeric (required).
status	Indicates if the application is available when RHF is initiated. Values may be YES or NO. Default is YES.
m	Maximum simultaneous connections the application may have. The maximum value is 127. Default is one.
x	Maximum number of simultaneously active copies of this application that are allowed. Maximum value is 127. Default is one.
s	Indicates if the application is a servicer, initated by RHF on request of a remote system. Values may be YES or NO. Default is NO.
a	Indicates if the application is started whenever RHF is initated or when the operator enables the application. Values may be YES or NO. Default is NO.
sys	Indicates if the application must be of system origin to perform a NETON. Values may be YES or NO. Default is YES.

RCFGEN automatically generates an application entry for MHF (Maintenance Host Facility):

```
APPL      NAME=MHF,ASTART=YES
```

If the installation does define MHF, the auto-start parameter must be enabled (ASTART=YES) for proper operation.

In determining the number of allowed connections and copies of an application, note that each NAD has a maximum of 127 active connections. This number is restricted to 35 during NAD controlware loading but may be increased by modifying the appropriate NAD controlware load parameters in BCLOAD.

The following APPL statements are an example of how to define an APPL table.

```
APPL    NAME=QTF,MXCONS=4,ASTART=YES
APPL    NAME=QTFS,MXCOPYS=4,SVR=YES
APPL    NAME=FTF,MXCOPYS=4
APPL    NAME=FTFS,MXCOPYS=8,SVR=YES
APPL    NAME=USRAP,MXCONS=6,ENABLED=NO
APPL    NAME=MLTF,ASTART=YES
APPL    NAME=ITF,MXCONS=2
```

NOTE

When defining the APPL statements you must follow certain restrictions for system-supplied applications QTF, QTFS, FTF, and FTFS. MXCONS should be set to one (default) for QTFS, FTF, and FTFS. MXCOPYS should be set to one (default) for QTF. MXCONS should not be set greater than four unless installation parameter MAXFILEXFR is increased for QTF/FIP. The SVR=YES parameter must be specified for QTFS and FTFS and must not be specified for QTF and FTF (either SVR=NO or default).

DEFINITION OF MAXIMUM NDRs ALLOWED FOR ALL NADs

To define the maximum NDRs allowed for all NADs, enter:

```
LNDR    MAXNDRS=nn
```

nn   Maximum number of PPs that may contain NDR at any one time, regardless of the number of NADs and NDRs allowed per NAD. Default is one.

DEBUG PARAMETER DEFINITION

To define the DEBUG parameter, enter:

```
DEBUG   TRACE=t
```

t — Values may be YES or NO. NO specifies that RHF trace is off. When RHF trace is off, queue entries freed by RHF are placed at the top of the empty queue and immediately reused. When NO is specified, RHF uses a slightly smaller amount of CP time. Default is NO.

YES specifies that RHF trace is on. When trace is on, queue entries are reused only after all queue entries ahead of it. Also, when trace is on, analysis of an RHF dump and resolution of the associated RHF problem may be facilitated.

UCP CHARGE DEFINITION FOR AN RHF CALL

This statement specifies the amount of system resources a UCP is charged for an RHF call. RHF distinguishes two different types of calls; those that require a large amount of processing time and those that require a small amount of processing time. RHF charges the UCP more for the former category of calls.

```
CHARGE   TYPE=n,CPA=cpa,CPB=cpb,IO=io,CMFL=cm,PP=pp
```

Parameter	Description
nn	Type of call for which the charge is being specified. Value may be 1 or 2. 1 requires small amount of RHF processing time 2 requires large amount of RHF processing time
cpa	CPA time to be charged (decimal milliseconds). Default is 2 milliseconds for a type 1 call, 10 milliseconds for a type 2 call.
cpb	CPB time to be charged (decimal milliseconds). Default is 0.
io	IO time to be charged (decimal milliseconds). Default is 0.
cm	CM field length (octal)/100B to be charged. Default is 10.
pp	PP time to be charged (decimal milliseconds). Default is 0.

A CHARGE statement is not required. You may, however, enter two CHARGE statements; one for type 1 calls, a second for type 2 calls.

VERIFICATION PROGRAM

Modify and run job PL1LV to verify the installation of RHF and applications MFLINK, FTFS, QTF, and QTFS. Comments within the job indicate the changes needed to suit the local hardware configuration.

After PL1LV has executed, the installation of the Interactive Transfer Facility (ITF) applicaton can be verified using the procedure described in the section titled Interactive Transfer Facility.

QTF CONFIGURATIONS REQUIREMENTS

You must create the appropriate configuration files before you can perform queued file transfers using RHF.

RHF Configuration File Directives

For each copy of QTF that you want RHF to start automatically on a NOS/BE host, you must include the following network configuration statement in the RCFGEN input file for that host:

```
APPL     NAME=QTF,MXCONS=4,ASTART=YES
```

Refer to the APPL statement under RHF Configuration Files in this section.

Each copy of QTF will establish at most four simultaneous connections. For most RHF configurations, a single copy of QTF should be sufficient.

To allow a NOS/BE host to receive queued files, you must include the following network configuration statement in the RCFGEN input file for that host:

```
APPL     NAME=QTFS,SVR=YES,MXCOPYS=n
```

where n is the maximum number of QTFS servers you want to have active. Each copy of QTFS services one connection.

Refer to the appropriate reference manuals for other RHF implementations for information on how to configure non-NOS/BE QTF and QTFS applications and connections.

QTF Procedure File

The QTF procedure file contains a set of default initialization commands. You can edit the commands in the procedure and capture the modified procedure to a new system file without rebuilding QTF, or you can include UPDATE modifications in the PL91I installation procedure. The QTF procedure is deck QTFPROC on the RHP (PL91) program library file.

Format of the call to the QTF procedure file is:

```
QTF,I=infile.
```

Parameter	Description
infile	File name from which initialization commands are to be processed. Default is to use the commands included in the QTF procedure file. QTF reads the initialization commands until end-of-record. Each command on the file is a separate line (equivalent to an L-display entry. Specifying I-infile is equivalent to entering the command INCLUDE,FILE=infile.

Procedure File Example

The following example shows some typical sets of initialization commands that you can include in the QTF procedure file. The example assumes you started with the default QTF procedure file.

```
.PROC,QTF*I, ...
     .
     .
     .
.DATA,XXDEFI
     .
     .
     .
.*
.* DEFAULT DIRECTIVES FOR NOS/BE VARIANT
.*
SCHED,MAXCONS=4.
SCHED,FSI6=32768.
CLASS,SC=A,FSI=1..4.
CLASS,SC=B,FSI=1..6.
CLASS,SC=C,FSI=1..6.
CLASS,SC=D,FSI=7,MAXIMUM=1.  AT MOST ONE LARGE FILE.
DISABLE,SC=D,PID=M03.  NEVER ALLOW LARGEST FILES TO M03.
.IF,(TIME.GT.0700.AND.(TIME.LT.1700),PRIME.
INCLUDE,F=PRIME.  MUST BE LAST DIRECTIVE
.ELSE,PRIME.
INCLUDE,F=OFF.    MUST BE LAST DIRECTIVE
.ENDIF,PRIME.
.DATA,PRIME
  . PRIME-HOURS PARAMETERS
  . ENTER "K.INCLUDE,F=PRIME." TO CHANGE NAM PARAMETERS
DISABLE,SC=D.   NO LARGE FILES DURING PRIME TIME.
ENABLE,SC=B,PID=M03.   PERMIT MEDIUM FILES TO M03.
ENABLE,SC=C,PID=M03.
.DATA,OFF
  . OFF-HOURS PARAMETERS
  . ENTER "K.INCLUDE,F=OFF." TO CHANGE NAM PARAMETERS
ENABLE,SC=D.    ALLOW LARGE FILES EXCEPT TO M03.
DISABLE,SC=B,PID=M03.  DISALLOW MEDIUM FILES TO M03.
DISABLE, SC=C,PID=M03.
```

The Interactive Transfer Facility is a multi-user job application program used by the Remote Host Facility allowing NOS/BE INTERCOM users to enter commands on an LCN-linked mainframe.

RELEASE MATERIALS

ITF is released on tape PL92. The structure of the release tape is as follows.

File 1 OLDPL

File 2 Nucleus absolutes

HARDWARE CONFIGURATION

ITF requires the same minimum configuration required for INTERCOM under NOS/BE plus a minimum of one 380-170 network access device (NAD).

INSTALLATION PROCEDURE

Obtain installation job decks PL92I and PL92E from the installation deck PL using the procedure outlined in section I, part 1.

INSTALLATION PARAMETERS

Any changes which cause the size of ITF to increase may require that you increase the field length for ITF in the MUJTBL for 1QP.

The format for all installation parameters is:

```
1     7     11             22
            DEF parameter  #value#;
```

MUJFETS — The maximum number of FETs ITF uses for terminal input and output. Default value is 3. The change deletion location is COMIDEF.17. The following is an example parameter change:

```
*DELETE    COMDEF.17
  DEF      MUJFETS        #5#;
```

MUJUSRS — The maximum number of simultaneous users allowed in ITF. Default value is 30. The change deletion location is COMIDEF.18. The following is an example of a parameter change.

```
*DELETE      COMIDEF.18
      DEF    MUJUSRS         #50#;
```

MUJTBL$L — The size of storage needed by ITF MUJ subroutine tables. Default value is 138. MUJTBL$L is equivalent to MUJFETS*6 + MUJUSRS*4. The change deletion location is COMIDEF.19. The following is an example of a parameter change.

```
*DELETE      COMIDEF.19
      DEF    MUJTBL$L        #230#;
```

TRMINLN/
TRMOTLN — Length of communication area between 1QP and ITF. Default values for both are 10. The change deletion locations are DEFCOM.5 (TRMINLN) and DEFCOM.6 (TRMOTLN). The following is an example of a parameter change.

```
*DELETE      DEFCOM.5
      DEF    TRMINLN         #12#;
*DELETE      DEFCOM.6
      DEF    TRMOTLN         #12#;
```

THRSHLD — Determines the minimum number of CP seconds accumulated before accounting is posted to the user. Default value is 2. The change deletion location is DEFCOM.2. As THRSHLD is set to smaller values, accounting is more accurate, overhead is increased, and THRSHLD has more meaning.

NACOUNT — Determines how frequently the accounting information for ITF is obtained from the system and distributed to active ITF users. NACOUNT is the maximum number of user switches performed on any given user before accounting is done. Accounting is always performed on user exit from ITF. Values must be greater than or equal to one. Default is 40. The change deletion location is DEFCOM.9.

INSTALLID — Specifies the LID to which remote mainframe connection is made. The terminal user is not prompted to enter a LID. The LID is three alphanumeric characters enclosed in quotes. Default entry is 0. The change deletion location is COMIDEF.21. The following is an example of a parameter change:

```
*DELETE      COMIDEF.21
      DEF    INSTALLED  #"MFT"#;
```

MAXTCN — Maximum number of users per remote mainframe connection. Default value is 30. The change deletion location is COMITBLS.20.

TBLLENG — Maximum number of allowed remote mainframes. Default value is 2. The change deletion location is COMITBLS.17.

MAXACN — Upper bound for connection number. MAXACN must be equal to TBLLENG. Default value is 2. The change deletion location is COMITBLS.19.

VERIFICATION PROCEDURE

The first step in verifying that ITF has been installed correctly is to execute PL1LV, the RHF verification program described in the section titled Remote Host Facility. After PL1LV has been completed successfully and while RHF is still active, bring up INTERCOM. Log in at a terminal, and enter MFINT.

The correct response to the MFINT command depends on the value assigned to parameter INSTALLID when ITF was installed.

If INSTALLID was not changed, the response should be:

    PLEASE ENTER REMOTE MF LID

Enter LBK. The response should be:

    ITF, CONNECTION UNAVAILABLE

This indicates that ITF was installed correctly.

If INSTALLID was changed to specify a CYBER 200 LID, the response should be:

    CYBER 200 LOGON banner

This indicates that ITF was installed correctly.

If the response is ITF, CONNECTION UNAVAILABLE or there is no response, activate RHF on the CYBER 200 and enter MFINT from the terminal to repeat the test.

NOS/VE DUAL-STATE PREPARATION 44

This chapter describes the steps you must perform to prepare the NOS/BE operating system for dual-state operations with NOS/VE version 1.2.1 and later releases. When you have completed the steps included in this chapter, refer to the NOS/VE Software Release Bulletin for instructions on how to create procedures that initiate NOS/VE deadstarts and how to install NOS/VE.

SOFTWARE REQUIREMENTS

The minimum software requirements for NOS/BE version 1.5 are sufficient to install NOS/VE. Use of the NOS/VE Interactive Facility (VEIAF) requires the installation of INTERCOM 5. The installation process assumes the presence of INTERCOM 5 and VEIAF; however, all interactive procedures can be executed in batch jobs.

BUILDING NOS/VE DUAL-STATE BINARIES ON NOS/BE

NOS/VE dual-state code can be rebuilt on NOS/BE by using the installed deck NVEBUILD from the DECKS program library. This deck will use as input the dual-state source library and any local code that the site has. This deck must be modified to change the IDs for the permanent file operations. Refer to the documentation included in the deck for further information.

The NVEBUILD deck assumes that the level of NOS/VE that you want to build is already installed on the system. NVEBUILD requires the use of files from that level of the NOS/VE installation.

The NVEBUILD deck does not rebuild the CYBIL part of NOS/VE dual-state, since CYBIL is a separate product and does not have to be installed for dual-state to operate. If you have installed CYBIL and you want to rebuild the CYBIL portions, the NVEBUILD deck can be modified to do so. Refer to the documentation in the NVEBUILD deck for further details.

STEP 1, ESTABLISH A NOS/VE INSTALLATION USER NAME

The first step in the installation process is to select a user name on NOS/BE under which you will install NOS/VE.

NOS/BE permanent files created during the NOS/VE installation or upgrade process are associated with a specific NOS/VE user name. All permanent files created during installation will have this user name as a permanent file ID.

The user name selected in this step should be used whenever a UN parameter is specified in an installation procedure call. This user name will be validated in step 7.

STEP 2, GENERATE NOS/BE CONFIGURATION CHANGES

Steps 2 through 6 generate and install NOS/BE configuration changes required to install NOS/VE.

The process of creating NOS/BE deadstart files tailored for individual sites is described in part II, chapter 1. In addition to the material contained in that chapter, the following items should be addressed in preparation for the generation of a new NOS/BE deadstart file to support dual-state configuration:

- NVE subsystem support
- Equipment status table (EST) definitions
- Central memory (CM) allocation

NVE SUBSYSTEM SUPPORT

The NOS/VE operating system appears to NOS/BE primarily as the NVE subsystem. To fully support the NVE subsystem, add the following to the CMR installation parameters (CMRIP):

```
*INSERT,CMRIP.1
 SSNVE    CEQU     SS.NVE        (Turns on NVE subsystem support)
 N.SBSYS  CEQU     5             (Sets the number of subsystems)
```

EST DEFINITIONS

Be sure that all equipment to be used by either NOS/BE or NOS/VE is defined in the equipment status table (EST). Define disk units to be used by NOS/VE as IDLE and disk channels to be used by NOS/VE as OFF.

CM ALLOCATION

NOS/BE deadstart tapes may be prepared with or without Unified Extended Memory (UEM) defined. In cases where UEM is defined, its use is optional at deadstart time. To use UEM, however, a system must have more than 8 megabytes of central memory. To use UEM, NOS/BE needs 2 megabytes of central memory (not counting the memory assigned to UEM). In addition, NOS/VE needs 6 megabytes. Thus, an 8-megabyte system would have no memory to spare for UEM.

Without UEM:

The amount of central memory allocated to NOS/BE is normally 2 megabytes (262,144 words), with the remainder assigned to NOS/VE. If that remainder is less than the minimum needed for NOS/VE, then the central memory allocated to NOS/BE can be automatically reduced to help increase the amount available to NOS/VE. For the parameter settings needed to make this automatic reduction, refer to Deadstart Installation Parameters in part II, chapter 1.

With UEM:

The amount of memory assigned to NOS/BE is the sum of the 2 megabytes (262,144 words) used as central memory and the amount defined as UEM by the Extended Memory Label which is constructed in CMR by the ECSLABEL and ECSPART macros, as described under Macros to Construct Extended Memory Label in part II, chapter 1. This amount can never be greater than 16 megabytes (2,097,152 words); the remaining amount is assigned to NOS/VE (it must be at least 6 megabytes).

In the following example, assume the mainframe has 12 megabytes (1,572,864 words) of memory. The example constructs an ECS label which defines 2 megabytes (1000000B words) of memory as UEM for NOS/BE. A NOS/BE system deadstarted with this label would then require a total of 4 megabytes of memory, 2 megabytes for CM and 2 megabytes for UEM, leaving the remainder of the memory for NOS/VE. This amount would be less than 8 megabytes, since some memory is used for the environment interface (EI) and the page table.

```
*INSERT CMR.2167
      ECSLABEL    1000
      ECSPART     NOSBE,1,300
      ECSPART     NOSBE,2,477
```

The ECS label defines 1000000B words (262,144 decimal words) in this example, with 300000B words (98,304 decimal words) as a direct access partition and 477000B words (163,328 decimal words) as a paged partition; both are assigned to the mainframe with the ID of NOSBE. The sum of the partitioned area is 1000B words (512 decimal words) less than the total amount of memory for UEM to accommodate the ECS label.

STEP 3, INSTALL THE NOS/VE RELEASE FILES

To install the NOS/VE release files, you must run the build job NVEINST from the DECKS program library. This build job must be modified to use the ID defined in step 1. The build job requires the NOS/BE dual-state tape for NOS/VE.

STEP 4, CREATE THE NOS/BE DEADSTART FILE

You must now generate a new NOS/BE deadstart file containing the changes you made in step 2 and the binaries and procedures from files NBEBINS and NVELIBB.

Part I of this manual describes the process of building a NOS/BE deadstart tape. You should be familiar with this process before attempting this step.

The standard NOS/BE installation decks have been modified to support installation of NOS/VE. These modifications include a new deck, NVEIE, to install the NOS/VE release files. The installation decks are contained on an UPDATE program library on file 3 of the NOS/BE batch corrective code (BCC) tape.

Submit the following job to obtain a listing of the installation decks and then save the DECKS old program library on a permanent file for later use.

```
DECKS,T100,PE1.
REQUEST(OLDPL,PE)              (Assigns BCC tape)
REQUEST(NEWPL,PF)
SKIPF(OLDPL,2,17)
UPDATE(F,N,*==,C=0,R,L=A7)
CATALOG(NEWPL,DECKS,ID=INSTALL)
6/7/8/9
```

You must generate an UPDATE modification set against the DECKS old program library (OLDPL) to create the NOS/BE build jobs. This modification set must include the following statement:

```
=DEFINE NOSVE
```

This statement causes the build jobs that are generated to include NOS/VE in the list of products being installed.

You must also include changes to the NVEIE job in the modification set. This replaces the username fields in the NVEIE job with the user name you defined in step 1. Execution of the NVEIE build job is dependent on the NOS/VE release files being available as permanent files and cataloged with the user name from step 1 as the ID.

STEP 5, REDEADSTART NOS/BE

Redeadstart NOS/BE using the deadstart file created in step 3.

STEP 6, ADD INFORMATION FOR PARTNER JOBS

Steps 6 and 7 make NOS/BE accounting changes for the NOS/VE partner jobs. Step 6 is required only if your site has local accounting requirements; if not, go to step 7.

Step 6 describes how to format the partner job template. The partner job template contains the accounting information your site has designed for NOS/BE. This accounting information is used by the NOS/VE Interim Remote Host Facility (IRHF) and Interstate Communication Facility (ICF) in creating NOS/BE job images. The partner job template contains a field that contains the NOS/BE job statement parameters and optional fields containing the NOS/VE family user accounting parameters. They must be in a format that can be processed by your site's NOS/BE batch job accounting utility. NOS/VE users have four accounting parameters: user name, password, account number, and project number (described in the NOS/VE Family Administration manual). These parameters may be specified in the NOS/BE partner job image according to the format of the partner job template. Examples of how to format the partner job template are shown at the end of this step.

The partner job template is a record that contains one or more of the following fields.

&JOB	This field will be replaced by the NOS/BE job statement parameters specified in the partner job template.
&USER	This field, if present, will be replaced by the NOS/VE user name.
&PASSWORD	This field, if present, will be replaced by the NOS/VE password.
&CHARGE	This field, if present, will be replaced by the NOS/VE account name.
&PROJECT	This field, if present, will be replaced by the NOS/VE project name.

The partner job template records are on the released procedures ACCFILE and RUNIRHF. ACCFILE and RUNIRHF are released on file NVELIBB and support the generation of a standard NOS/BE job control statement. The procedure ACCFILE is executed by the NOS/VE deadstart procedure from the NVELIB library during the deadstart process. When ACCFILE is executed, two partner job templates are created as local files named RHACCNT and ICACCNT. The RHACCNT file is used for all GET_FILE and REPLACE_FILE partner jobs. The ICACCNT file is used for all Interstate Communication partner jobs (like CREIC partner jobs). The RUNIRHF procedure is executed whenever the IRHF170 job is brought up. When RUNIRHF is executed, a partner job template is created as a local file named PRACCNT. The PRACCNT file is used for the PRINT_FILE command that uses the DSRP parameter. These partner job templates are then available to the IRHF and ICF when NOS/BE jobs are submitted.

To access and modify the released versions of ACCFILE and RUNIRHF, execute the following commands:

```
ATTACH,NVELIB,ID=username,MR=1.
LIBRARY,NVELIB.
GETPROC,ACCFILE,NVELIBB,UN=username.
GETPROC,RUNIRHF,NVELIBB,UN=username.
```

where username is the user name you selected in step 1.

These commands get procedures ACCFILE and RUNIRHF from NVELIBB and copies them to local files named ACCFILE and RUNIRHF. You must modify the partner job template records. In procedure ACCFILE, they are the RHACCNT and ICACCNT data records; in procedure RUNIRHF, it is the PRACCNT data record. Modify these records to reflect your site's accounting requirements.

```
.PROC,ACCFILE*I,
.
.HELP
THIS PROCEDURE CREATES THE PARTNER JOB TEMPLATE FILE USED
BY THE REMOTE HOST AND INTERSTATE COMMUNICATION EXECUTIVES
TO GENERATE PROPER ACCOUNTING INFORMATION FOR PARTNER JOBS.
THE CARDS IN THE ACCOUNT DATA SECTION MAKE UP THE PARTNER
JOB FILE IMAGE.

.ENDHELP
REWIND,ICACCNT.
REWIND,RHACCNT.
REVERT. ICACCNT AND RHACCNT FILES CREATED.
.DATA,ICACCNT.
&JOB
.DATA,RHACCNT.
&JOB
```

ACCFILE and RUNIRHF can then be replaced on NVELIBB by the following commands:

```
REPPROC,ACCFILE,NVELIBB,UN=username.
REPPROC,RUNIRHF,NVELIBB,UN=username.
```

where username is the user name you selected in step 1.

NOTE

The CCL procedure RUNIRHF, which is on NVELIBB, is used to initiate the NOS/VE remote host job. If your site requires accounting information to be added to the remote host job record contained in RUNIRHF, you must do so before starting step 7. To add the accounting information, enter the commands listed in step 6 for accessing and modifying the release version of ACCFILE, substituting RUNIRHF for ACCFILE.

The following examples demonstrate how you might replace the DATA section of the ACCFILE procedure to describe the accounting parameters for your site.

Example 1:

If your site does not require batch job accounting parameters, no changes need to be made to the released version. It remains:

```
&JOB
```

Example 2:

If your site requires the user name, account number, and project number on the job control statement, replace &JOB with:

```
&JOB      &USER,&CHARGE,&PROJECT
```

Example 3:

If your site requires the user name, password, and account and project number as input to a local account program, replace &JOB with:

```
&JOB
ACCOUNT(&USER,&PASSWORD,&CHARGE,&PROJECT)
```

STEP 7, VALIDATE USERS ON NOS/BE

To be an interactive user of NOS/VE, you must be a validated user of NOS/BE INTERCOM 5. To log in interactively to NOS/VE, you must first log in to INTERCOM 5 on NOS/BE and then log in to NOS/VE by executing an INTERCOM 5 multiuser job named VEIAF. You must use the PASSWRD utility (used to validate NOS/BE INTERCOM 5 users) to validate yourself as a NOS/VE user as well. A complete description of the PASSWRD utility can be found in part II, chapter 28, INTERCOM Version 5. The NOS/VE user name you established in step 1 must be validated in the INTERCOM password file.

You can use the following job to validate a NOS/VE user:

```
JOBCARD.
PASSWRD.
7/8/9
OLD
ADD U=beuser,P=password,W=veuser
6/7/8/9
```

U The NOS/BE user name of up to 10 alphanumeric characters. A user name must be specified for restricted passwords; it must be left blank or omitted for unrestricted passwords.

P The NOS/BE password of up to 10 alphanumeric characters. A password is required.

W The NOS/VE user name to be associated with the specified NOS/BE user name and password. It must be from 1 to 7 alphanumeric characters. The W parameter must be specified to permit this NOS/BE user to log in to NOS/VE.

PART III

CROSS-REFERENCE OF INSTALLATION PARAMETERS 1

****PL1A****

****
IPARAMS
****

IP.ACNT	1AJ					
IP.BOVF	BBJ	1IB				
IP.BRCL	CPSCH					
IP.CMBL	REQEBUF	GETRAND	STORE	FLUSHST		
IP.CMPID	CMR					
IP.BCFAP	CLRCEM	ECSUB	CEM			
IP.CMU	CED					
IP.CPLM	7AJ	RPV				
IP.C63	1RN	1MT	1P2	1P4	1P3	1P1
	1WI	1RV	1NO	1N2	1N3	1R2
	1R3	1TF	1WS	1NW	8T3	2TB
	1CR	1CT	1CS	1RT	1RS	1NR
	1W9	1C9	1R9	4LB	4LC	
IP.CSET	CED	1RN	1IT	2IR	3II	2IW
	2IA	2IC	2ID	2IP	2IT	2IX
	1MT	1P2	1P4	1P3	1P1	1WI
	1RV	1NO	1N2	1N3	1R2	1R3
	1TF	1WS	1NW	8T3	2TB	1CR
	1CT	1CS	1RT	1RS	1NR	1W9
	1C9	1R9	4LB	7T1	7T2	4LC
	1LC					
IP.C176	CED	CMR	CPMTR	SEGLINK	MTR	
	DSD	1SC				
IP.DCT	CMR					
IP.DECR	CCP	CSWP				
IP.ECFL	CMR					
IP.DBAL	1IB	1AB				
IP.ECSB	CMRTEXT	CED	IRCP	STL	CMR	CPMTR
	EXBOOT	MTR	DSD	1AJ		
	7EC	1EJ	3DO	1RN	1SP	MEM
	REQ	3RQ	QAF	QAC	1IB	1SI
	1SO	CEM	TDS			
IP.ENAP	1AJ	1MH	1SI			
IP.FTHRL	CMR					
IP.IACES	1AJ					
IP.ILCMD	DSD					
IP.INT5	PPTEXT	SCPTEXT	CMR	DSD		
IP.IOLM	7AJ	RPV				
IP.IQD	1RN					
IP.IQPW	1IB					
IP.IRCL	CPSCH					
IP.IUSID	CMR					
IP.LINK	DSD					
IP.LVF	1EJ	4EJ	1RN	1IB	1TJ	2VJ
	6RD					
IP.MCPU	CMRTEXT	CED	CMR	CPMTR	SETST	CPUST
	USERMOD	TIMSEG	SSCSEG	SSFSEG	CPSS	PACKAGE

	LINK	RESCH	UPM	CCP	SYSIDLE	WOR
	CMDATA					
	RAGET	MTR				
	DSD	7AJ	4EJ	1SP	DIS	1IB
	1SI	1SO	1TJ	2VJ	RPV	1SQ
IP.MECS	CMR	CPSS	PACKAGE	CPECSM		
	MTR	DSD	4EJ	DIS	MEM	
	1IB	1SI	1SO	1TJ	2VJ	
IP.MMS	1AJ					
IP.MPPU	CMR	SPRBMGR	MTR			
IP.MPR	4EJ	1TJ	2VJ			
IP.MSCT	1IB					
IP.MSLM	7AJ	RPV				
IP.MTL	4EJ	1TJ	2VJ			
IP.MXTIN	UPM					
IP.MXQT	CSWP					
IP.NDFS	1EJ					
IP.NJFL	BBJ					
IP.OPRI	6RD					
IP.OQD	1RN					
IP.POSFL	MTR	1SO				
IP.PPS2X	CED					
IP.RBINT	1RN					
IP.ROFL	UPM	1SO				
IP.SCHDE	IRCP	DSD	1EJ	4EJ	3RQ	3MN
	1CL	3IC	3IE	3IR	3IW	1IB
	1TJ	2VJ	1TS	1LC		
IP.SFL	DSD	1AJ	4EJ	1SI	1TJ	2VJ
	STS	1AB				
IP.SIDLE	CMR	CPMTR	RESCH	CPSCH	SYSIDLE	CMRDIR
	CMDATA	MTR				
	DSD	1SC	TDS			
IP.SIOL	4EJ	1TJ	2VJ			
IP.SLIB	CMR					
IP.SMS	1IB	1SI				
IP.SPR	4EJ	1TJ	2VJ	6RD		
IP.SPT	IRCP	CMR	CPMTR	CPSM	CPSPM	CPSCH
	BBJ					
	1SO					
IP.SRMS	CED	STL	1RN			
IP.STEX	DSD					
IP.STL	4EJ	1TJ	2VJ			
IP.SYSE	CMR					
IP.TCPUB	CPMTR	TIMSEG	RESCH			
IP.TOVF	BBJ	1IB				
IP.TYPE	LINK	RESCH	CPSM	SCHRES		
IP.VER	CMR					
IP.XJ	CPMTR	CPUST	RESCH	USERR	SCHRES	SEGPAR
	RAGET					
	CACT	MTR	DSD			
IP.YMD	IRCP	DSD				
IP.819	IPTEXT	CMRTEXT	CED	IRCP	STL	CMR
	CPSPM	CMRDIR	MTR	DSD	CIO	
	1AJ	7EC	1EJ	3DO	1S5	1RN
	1SP	DIS	MEM	REQ	3RQ	ACE
	1RP	1CL	1TR	1TO	6WM	1MH
	1IT	2IW	2IA	2IB	2IC	2ID

	2IL	2IO	2IP	2IT	2IX	3IC
	3IE	3IF	3IJ	3IL	3IM	3IN
	3IO	3IR	3IV	3IW	1MT	SSH
	QAF	QAC	1P2	1P4	1P3	1P1
	1WI	1RV	1NO	1N2	1N3	1R2
	1R3	1TF	1WS	1NW	8T3	2TB
	1CR	1CT	1CS	1RT	1RS	1NR
	1W9	1C9	1R9	1SC	1IB	1SI
	1SO	1TJ	2VJ	4LB	4LC	6RD
	CEM	1TS	RPV	SSC	STS	TDS
	1LC	1AB	REQUEST	LABEL		
MODEL	IPTEXT	CMRTEXT	CED	IRCP	STL	CMR
	MTR	DSD	CIO	1AJ	1EJ	3DO
	1S5	1RN	1SP	DIS	MEM	REQ
	3RQ	ACE	1RP	1CL	1TR	1TO
	6WM	1MH	1IT	2IW	2IA	2IB
	2IC	2ID	2IL	2IO	2IP	2IT
	2IX	3IC	3IE	3IF	3IJ	3IL
	3IM	3IN	3IO	3IR	3IV	3IW
	1MT	SSH	QAF	QAC	1P2	1P4
	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW
	8T3	2TB	1CR	1CT	1CS	1RT
	1RS	1NR	1W9	1C9	1R9	1SC
	1IB	1SI	1SO	1TJ	2VJ	4LB
	4LC	6RD	CEM	1TS	RPV	SSC
	STS	TDS	1LC	1AB	REQUEST	LABEL
OS.NAME	IPTEXT	CMRTEXT	CED	IRCP	STL	CMR
	MTR	DSD	CIO	1AJ	1EJ	3DO
	1S5	1RN	1SP	DIS	MEM	REQ
	3RQ	ACE	1RP	1CL	1TR	1TO
	6WM	1MH	1IT	2IW	2IA	2IB
	2IC	2ID	2IL	2IO	2IP	2IT
	2IX	3IC	3IE	3IF	3IJ	3IL
	3IM	3IN	3IO	3IR	3IV	3IW
	1MT	SSH	QAF	QAC	1P2	1P4
	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW
	8T3	2TB	1CR	1CT	1CS	1RT
	1RS	1NR	1W9	1C9	1R9	1SC
	1IB	1SI	1SO	1TJ	2VJ	4LB
	4LC	6RD	CEM	1TS	RPV	SSC
	STS	TDS	1LC	1AB	REQUEST	LABEL
OS.VER	IPTEXT	CMRTEXT	CED	IRCP	STL	CMR
	MTR	DSD	CIO	1AJ	1EJ	3DO
	1S5	1RN	1SP	DIS	MEM	REQ
	3RQ	ACE	1RP	1CL	1TR	1TO
	6WM	1MH	1IT	2IW	2IA	2IB
	2IC	2ID	2IL	2IO	2IP	2IT
	2IX	3IC	3IE	3IF	3IJ	3IL
	3IM	3IN	3IO	3IR	3IV	3IW
	1MT	SSH	QAF	QAC	1P2	1P4
	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW
	8T3	2TB	1CR	1CT	1CS	1RT
	1RS	1NR	1W9	1C9	1R9	1SC
	1IB	1SI	1SO	1TJ	2VJ	4LB
	4LC	6RD	CEM	1TS	RPV	SSC

	STS	TDS	1LC	1AB	REQUEST	LABEL
PR.BATCH	IPTEXT	CMRTEXT	CED	IRCP	STL	CMR
	MTR	DSD	CIO	1AJ	1EJ	3DO
	1S5	1RN	1SP	DIS	MEM	REQ
	3RQ	ACE	1RP	1CL	1TR	1TO
	6WM	1MH	1IT	2IW	2IA	2IB
	2IC	2ID	2IL	2IO	2IP	2IT
	2IX	3IC	3IE	3IF	3IJ	3IL
	3IM	3IN	3IO	3IR	3IV	3IW
	1MT	SSH	QAF	QAC	1P2	1P4
	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW
	8T3	2TB	1CR	1CT	1CS	1RT
	1RS	1NR	1W9	1C9	1R9	1SC
	1IB	1SI	1SO	1TJ	2VJ	4LB
	4LC	6RD	CEM	1TS	RPV	SSC
	STS	TDS	1LC	1AB	REQUEST	LABEL
PR.NUMB	CMR					
PR.INT	IPTEXT	CMRTEXT	CED	IRCP	STL	CMR
	MTR	DSD	CIO	1AJ	1EJ	3DO
	1S5	1RN	1SP	DIS	MEM	REQ
	3RQ	ACE	1RP	1CL	1TR	1TO
	6WM	1MH	1IT	2IW	2IA	2IB
	2IC	2ID	2IL	2IO	2IP	2IT
	2IX	3IC	3IE	3IF	3IJ	3IL
	3IM	3IN	3IO	3IR	3IV	3IW
	1MT	SSH	QAF	QAC	1P2	1P4
	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW
	8T3	2TB	1CR	1CT	1CS	1RT
	1RS	1NR	1W9	1C9	1R9	1SC
	1IB	1SI	1SO	1TJ	2VJ	4LB
	4LC	6RD	CEM	1TS	RPV	SSC
	STS	TDS	1LC	1AB	REQUEST	LABEL
PR.SCP	IPTEXT	CMRTEXT	CED	IRCP	STL	CMR
	MTR	DSD	CIO	1AJ	1EJ	3DO
	1S5	1RN	1SP	DIS	MEM	REQ
	3RQ	ACE	1RP	1CL	1TR	1TO
	6WM	1MH	1IT	2IW	2IA	2IB
	2IC	2ID	2IL	2IO	2IP	2IT
	2IX	3IC	3IE	3IF	3IJ	3IL
	3IM	3IN	3IO	3IR	3IV	3IW
	1MT	SSH	QAF	QAC	1P2	1P4
	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW
	8T3	2TB	1CR	1CT	1CS	1RT
	1RS	1NR	1W9	1C9	1R9	1SC
	1IB	1SI	1SO	1TJ	2VJ	4LB
	4LC	6RD	CEM	1TS	RPV	SSC
	STS	TDS	1LC	1AB	REQUEST	LABEL
PR.SYS	CMR	CPUST				
********						
CIOCOM						
********						
IP.D7DN	CED					
IP.NBCD	1EJ	3MN	8T3	6LC	6L3	1TS
IP.NBRK	3MN					
IP.NDEN	CMRTEXT	CED	CMR	DSD	CIO	1EJ

	4EJ	3DO	4DO	1RN	4ES	REQ
	3RQ	3MN	1RP	1CL	2TC	1TR
	1TO	1OP	1BT	6WM	1MF	1MH
	1IT	2IW	2IA	2IB	2IC	2ID
	2IL	2IO	2IP	2IT	2IX	3IC
	3IE	3IF	3IJ	3IL	3IM	3IN
	3IO	3IR	3IV	3IW	1MT	1P2
	1P4	1P3	1P1	1WI	1RV	1NO
	1N2	1N3	1R2	1R3	1TF	1WS
	1NW	8T3	2TB	1CR	1CT	1CS
	1RT	1RS	1NR	1W9	1C9	1R9
	1IB	1TJ	2VJ	4LB	4LC	1TS
	REQUEST	LABEL				
IP.NOISE	1IT	3II	1MT	1P1	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW
	2TB	1RS	1NR			
IP.NOIS9	1IT	3II	1P1	1NO	1N2	1N3
	1R2	1R3	1TF	1NW	2TB	1NR
IP.NTCN	1RN	1MT	1P2	1P4	1P3	1P1
	1WI	1RV	1NO	1N2	1N3	1R2
	1R3	1TF	1WS	1NW	8T3	2TB
	1CR	1CT	1CS	1RT	1RS	1NR
	1W9	1C9	1R9	4LB	4LC	
IP.PCL	1RN	1MT	1P2	1P4	1P3	1P1
	1WI	1RV	1NO	1N2	1N3	1R2
	1R3	1TF	1WS	1NW	8T3	2TB
	1CR	1CT	1CS	1RT	1RS	1NR
	1W9	1C9	1R9	4LB	4LC	
IP.PTCN	1RN	1MT	1P2	1P4	1P3	1P1
	1WI	1RV	1NO	1N2	1N3	1R2
	1R3	1TF	1WS	1NW	8T3	2TB
	1CR	1CT	1CS	1RT	1RS	1NR
	1W9	1C9	1R9	4LB	6LC	4LC
IP.RCYC	2IO	3IL	4LB	4LC	LABEL	
IP.RPE1	1RN	1MT	1P2	1P4	1P3	1P1
	1WI	1RV	1NO	1N2	1N3	1R2
	1R3	1TF	1WS	1NW	8T3	2TB
	1CR	1CT	1CS	1RT	1RS	1NR
	1W9	1C9	1R9	4LB	4LC	
IP.RPE2	1RN	1MT	1P2	1P4	1P3	1P1
	1WI	1RV	1NO	1N2	1N3	1R2
	1R3	1TF	1WS	1NW	8T3	2TB
	1CR	1CT	1CS	1RT	1RS	1NR
	1W9	1C9	1R9	4LB	4LC	
IP.TDEN	CMRTEXT	CED	CMR	DSD	CIO	1EJ
	3DO	4DO	1RN	4ES	REQ	3RQ
	3MN	1RP	1CL	2TC	1TR	1TO
	1OP	1BT	6WM	1MF	1MH	1IT
	2IW	2IA	2IB	2IC	2ID	2IL
	2IO	2IP	2IT	2IX	3IC	3IE
	3IF	3IJ	3IL	3IM	3IN	3IO
	3IR	3IV	3IW	1MT	1P2	1P4
	1P3	1P1	1WI	1RV	1NO	1N2
	1N3	1R2	1R3	1TF	1WS	1NW
	8T3	2TB	1CR	1CT	1CS	1RT
	1RS	1NR	1W9	1C9	1R9	1IB
	1TJ	2VJ	4LB	4LC	1TS	REQUEST
	LABEL					

IP.TSG	CMRTEXT	CED	CMR	DSD	CIO	1EJ
	3DO	4DO	1RN	4ES	REQ	1RP
	1CL	2TC	1TR	1TO	1OP	1BT
	6WM	1MF	1MH	1IT	2IW	2IA
	2IB	2IC	2ID	2IL	2IO	2IP
	2IT	2IX	3IC	3IE	3IF	3IJ
	3IL	3IM	3IN	3IO	3IR	3IV
	3IW	1MT	1P2	1P4	1P3	1P1
	1WI	1RV	1NO	1N2	1N3	1R2
	1R3	1TF	1WS	1NW	8T3	2TB
	1CR	1CT	1CS	1RT	1RS	1NR
	1W9	1C9	1R9	1IB	1TJ	2VJ
	4LB	4LC	1TS	REQUEST	LABEL	
IP.WEC	REQUEST	LABEL				
********						
ECSOM						
********						
IP.CMBL	REQEBUF	GETRAND	STORE	FLUSHST		
IP.CYSTP	DRVS					
IP.EBUF	IRCP					

IPARAMS SORTED BY ROUTINE

ACE	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP					
BBJ	IP.BOVF	IP.NJFL	IP.SPT	IP.TOVF		
CACT	IP.XJ					
CCP	IP.DECR	IP.MCPU				
CED	IP.CMU	IP.CSET	IP.C176	IP.ECSB	IP.MCPU	IP.PPS2X
	IP.SRMS	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.D7DN	IP.NDEN	IP.TDEN	IP.TSG
CEM	IP.BCFAP	IP.ECSB	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
CIO	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
CLRCEM	IP.BCFAP					
CMDATA	IP.MCPU	IP.SIDLE				
CMR	IP.CMPID	IP.C176	IP.DCT	IP.ECFL	IP.ECSB	IP.FTHRL
	IP.INT5	IP.IUSID	IP.MCPU	IP.MECS	IP.MPPU	IP.SIDLE
	IP.SLIB	IP.SPT	IP.SYSE	IP.VER	IP.819	MODEL
	OS.NAME	OS.VER	PR.BATCH	PR.NUMB	PR.INT	PR.SCP
	PR.SYS	IP.NDEN	IP.TDEN	IP.TSG		
CMRDIR	IP.SIDLE	IP.819				
CMRTEXT	IP.ECSB	IP.MCPU	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG
CPECSM	IP.MECS					
CPMTR	IP.C176	IP.ECSB	IP.MCPU			
	IP.SIDLE	IP.SPT	IP.TCPUB	IP.TCPUB:		
	IP.XJ					
CPSCH	IP.BRCL	IP.IRCL	IP.SIDLE	IP.SPT		
CPSM	IP.SPT	IP.TYPE				
CPSPM	IP.SPT	IP.819				
CPSS	IP.MCPU	IP.MECS				
CPUST	IP.MCPU	IP.XJ	PR.SYS			
CSWP	IP.DECR	IP.MXQT				
DIS	IP.MCPU	IP.MECS	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
DRVS	IP.CYSTP					

DSD	IP.C176	IP.ECSB	IP.ILCMB	IP.INT5	IP.LINK	IP.MCPU
	IP.MECS	IP.SCHDE	IP.SFL	IP.SIDLE	IP.STEX	IP.XJ
	IP.YMD	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
ECSUB	IP.BCFAP					
EXBOOT	IP.ECSB					
FLUSHST	IP.CMBL					
GETRAND	IP.CMBL					
IPTEXT	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP					
IRCP	IP.ECSB	IP.SCHDE	IP.SPT	IP.YMD	IP.819	MODEL
	OS.NAME	OS.VER	PR.BATCH	PR.INT	PR.SCP	IP.EBUF
LABEL	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.RCYC	IP.TDEN	IP.TSG	IP.WEC
LINK	IP.MCPU	IP.TYPE				
MEM	IP.ECSB	IP.MECS	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
MTR	IP.C176	IP.ECSB	IP.MCPU	IP.MECS	IP.MPPU	IP.POSFL
	IP.SIDLE	IP.XJ	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
PACKAGE	IP.MCPU	IP.MECS				
PPTEXT	IP.INT5					
QAC	IP.ECSB	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP				
QAF	IP.ECSB	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP				
RAGET	IP.MCPU	IP.XJ				
REQ	IP.ECSB	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
REQEBUF	IP.CMBL					
REQUEST	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	IP.WEC	
RESCH	IP.MCPU	IP.SIDLE	IP.TCPUB	IP.TCPUB:		
	IP.TYPE	IP.XJ				
RPV	IP.CPLM	IP.IOLM	IP.MCPU	IP.MSLM	IP.819	MODEL
	OS.NAME	OS.VER	PR.BATCH	PR.INT	PR.SCP	
SCHRES	IP.TYPE	IP.XJ				
SCPTEXT	IP.INT5					
SEGLINK	IP.C176					
SEGPAR	IP.XJ					
SETST	IP.MCPU					
SPRBMGR	IP.MPPU					
SSC	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP					
SSCSEG	IP.MCPU					
SSFSEG	IP.MCPU					
SSH	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP					
STL	IP.ECSB	IP.SRMS	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
STORE	IP.CMBL					
STS	IP.SFL	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP				
SYSIDLE	IP.MCPU	IP.SIDLE				
TDS	IP.ECSB	IP.SIDLE	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
TIMSEG	IP.MCPU	IP.TCPUB				
UPM	IP.MCPU	IP.MXTIN	IP.ROFL			

USERMOD	IP.MCPU					
USERR	IP.XJ					
WOR	IP.MCPU					
1AB	IP.DBAL	IP.SFL	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
1AJ	IP.ACNT	IP.ECSB	IP.ENAP	IP.IACES	IP.MMS	IP.SFL
	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP					
1BT	IP.NDEN	IP.TDEN	IP.TSG			
1CL	IP.SCHDE	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
1CR	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1CS	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1CT	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1C9	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1EJ	IP.ECSB	IP.LVF	IP.NDFS	IP.SCHDE	IP.819	MODEL
	OS.NAME	OS.VER	PR.BATCH	PR.INT	PR.SCP	IP.NBCD
	IP.NDEN	IP.TDEN	IP.TSG			
1IB	IP.BOVF	IP.DBAL	IP.ECSB	IP.IQPW	IP.LVF	IP.MCPU
	IP.MECS	IP.MSCT	IP.SCHDE	IP.SMS	IP.TOVF	IP.819
	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT	PR.SCP
	IP.NDEN	IP.TDEN	IP.TSG			
1IT	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9	IP.TDEN
	IP.TSG					
1LC	IP.CSET	IP.SCHDE	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
1MF	IP.NDEN	IP.TDEN	IP.TSG			
1MH	IP.ENAP	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
1MT	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NTCN
	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG
1NO	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
1NR	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
1NW	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
1N2	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
1N3	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9

	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
10P	IP.NDEN	IP.TDEN	IP.TSG			
1P1	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
1P2	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1P3	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1P4	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1RN	IP.C63	IP.CSET	IP.ECSB	IP.IQD	IP.LVF	IP.OQD
	IP.RBINT	IP.SRMS	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1RP	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
1RS	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NTCN
	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG
1RT	IP.C63	IPCSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1RV	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1R2	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
1R3	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
1R9	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1SC	IP.C176	IP.SIDLE	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
1SI	IP.ECSB	IP.ENAP	IP.MCPU	IP.MECS	IP.SFL	IP.SMS
	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP					
1SO	IP.ECSB	IP.MCPU	IP.MECS	IP.POSFL	IP.ROFL	IP.SPT
	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP					
1SP	IP.ECSB	IP.MCPU	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP			
1S5	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP					
1TF	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
1TJ	IP.LVF	IP.MCPU	IP.MECS	IP.MPR	IP.MTL	IP.SCHDE

	IP.SFL	IP.SIOL	IP.SPR	IP.STL	IP.819	MODEL
	OS.NAME	OS.VER	PR.BATCH	PR.INT	PR.SCP	IP.NDEN
	IP.TDEN	IP.TSG				
ITO	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
1TR	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
1TS	IP.SCHDE	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NBCD	IP.NDEN	IP.TDEN	IP.TSG
1WI	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
1WS	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NTCN
	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG
1W9	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG	
2IA	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
2IB	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
2IC	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
2ID	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
2IL	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
2IO	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.RCYC	IP.TDEN	IP.TSG	
2IP	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
2IR	IP.CSET					
2IT	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
2IW	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
2IX	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
2TB	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NOISE	IP.NOIS9
	IP.NTCN	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN
	IP.TSG					
2TC	IP.NDEN	IP.TDEN	IP.TSG			
2VJ	IP.LVF	IP.MCPU	IP.MECS	IP.MPR	IP.MTL	IP.SCHDE
	IP.SFL	IP.SIOL	IP.SPR	IP.STL	IP.819	MODEL
	OS.NAME	OS.VER	PR.BATCH	PR.INT	PR.SCP	IP.NDEN
	IP.TDEN	IP.TSG				
3DO	IP.ECSB	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
3IC	IP.SCHDE	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
3IE	IP.SCHDE	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
3IF	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
3II	IP.CSET	IP.NOISE	IP.NOIS9			
3IJ	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT

	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
3IL	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.RCYC	IP.TDEN	IP.TSG	
3IM	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
3IN	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
3IO	IP.819	MODEL	OS.NAME	OSVER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
3IR	IP.SCHDE	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
3IV	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
3IW	IP.SCHDE	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH
	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG	
3MN	IP.SCHDE	IP.NBCD	IP.NBRK	IP.NDEN	IP.TDEN	
3RQ	IP.ECSB	IP.SCHDE	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.TDEN	
4DO	IP.NDEN	IP.TDEN	IP.TSG			
4EJ	IP.LVF	IP.MCPU	IP.MECS	IP.MPR	IP.MTL	IP.SCHDE
	IP.SFL	IP.SIOL	IP.SPR	IP.STL	IP.NDEN	
4ES	IP.NDEN	IP.TDEN	IP.TSG			
4LB	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RCYC	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG
4LC	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NDEN	IP.NTCN	IP.PCL
	IP.PTCN	IP.RCYC	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG
6LC	IP.NBCD	IP.PTCN				
6L3	IP.NBCD					
6RD	IP.LVF	IP.OPRI	IP.SPR	IP.819	MODEL	OS.NAME
	OS.VER	PR.BATCH	PR.INT	PR.SCP		
6WM	IP.819	MODEL	OS.NAME	OS.VER	PR.BATCH	PR.INT
	PR.SCP	IP.NDEN	IP.TDEN	IP.TSG		
7AJ	IP.CPLM	IP.IOLM	IP.MCPU	IP.MSLM		
7EC	IP.ECSB	IP.819				
7T1	IP.CSET					
7T2	IP.CSET					
8T3	IP.C63	IP.CSET	IP.819	MODEL	OS.NAME	OS.VER
	PR.BATCH	PR.INT	PR.SCP	IP.NBCD	IP.NDEN	IP.NTCN
	IP.PCL	IP.PTCN	IP.RPE1	IP.RPE2	IP.TDEN	IP.TSG

****PL1B****

****
IPARAMS
****

IP.ARCH	PFA	PFP				
IP.CP	1IS					
IP.CR	1IR	2IS	3IS			
IP.CSET	1IR	1IS	3IS	4IS	CVL	
IP.ECSB	LDD	LDQ	LDV	LDW		
IP.IACES	GCC					
IP.INT5	RWE					
IP.LVF	1PF					
IP.MSCT	MSG					
IP.PFRP	1PC	PFC	1QF	SPF	1FC	PFA
	GPF	LPF	PFP	PFE	PFR	PFS
	OUX					

IP.RM	MNT	RELABEL	LABELMS			
IP.SRMS	MNT					
IP.TYPE	CVL					
IP.US	1PC	PFC	1QF	SPF	1FC	PFA
	GPF	LPF	PFP	PFE	PFR	PFS
	OUX					
IP.UP	PFA					
IP.YMD	1GM	DAYSWAP				
OS.ID	LOADO	LOADO3				
PR.BATCH	HDS					

IPARAMS SORTED BY ROUTINE

CVL	IP.CSET	IP.TYPE		
DAYSWAP	IP.YMD			
GCC	IP.IACES			
GPF	IP.PFRP	IP.US		
HDS	PR.BATCH			
LABELMS	IP.RM			
LDD	IP.ECSB			
LDQ	IP.ECSB			
LDV	IP.ECSB			
LDW	IP.ECSB			
LOADO	OS.ID			
LOADO3	OS.ID			
LPF	IP.PFRP	IP.US		
MNT	IP.RM	IP.SRMS		
MSG	IP.MSCT			
OUX	IP.PFRP	IP.US		
PFA	IP.ARCH	IP.PFRP	IP.US	IP.UP
PFC	IP.PFRP	IP.US		
PFE	IP.PFRP	IP.US		
PFP	IP.ARCH	IP.PFRP	IP.US	
PFR	IP.PFRP	IP.US		
PFS	IP.PFRP	IP.US		
RELABEL	IP.RM			
RWE	IP.INT5			
SPF	IP.PFRP	IP.US		
1FC	IP.PFRP	IP.US		
1GM	IP.YMD			
1IR	IP.CR	IP.CSET		
1IS	IP.CP	IP.CSET		
1PC	IP.PFRP	IP.US		
1PF	IP.LVF			
1QF	IP.PFRP	IP.US		
2IS	IP.CR			
3IS	IP.CR	IP.CSET		
4IS	IP.CSET			

****UPDATE****

****
IPARAMS
****

IP.C63	UPDATE		
IP.PD	ITEMIZE		
MODEL	UPDATE		
OS.NAME	COPYL	ITEMIZE	UPDATE
OS.VER	COPYL	ITEMIZE	UPDATE

```
                         IPARAMS SORTED BY ROUTINE

  COPYL                  OS.NAME     OS.VER
  ITEMIZE                IP.PD       OS.NAME     OS.VER
  UPDATE                 IP.C63      MODEL       OS.NAME    OS.VER
****CYBER LOADER****

********
  IPARAMS
********
  IP.IACES               LOADER      LOADZ       LOADU
  IP.MECS                LOADER      LOADG       LOADS      LOADZ       LOADM       LOADU
                         LOADUM      UCLOAD
  IP.PD                  LOADER      LOADC       LOADS      LOADZ       LOADM       LOADU
                         LOADUC      LOADUM
  IP.PS                  LOADER      LOADC       LOADS      LOADZ       LOADM       LOADU
                         LOADUC      LOADUM
  OS.ID                  LOADER      LOADZ       LOADU

                         IPARAMS SORTED BY ROUTINE

  LOADC                  IP.PD       IP.PS
  LOADER                 IP.IACES    IP.MECS     IP.PD      IP.PS       OS.ID
  LOADG                  IP.MECS
  LOADM                  IP.MECS     IP.PD       IP.PS
  LOADS                  IP.MECS     IP.PD       IP.PS
  LOADU                  IP.IACES    IP.MECS     IP.PD      IP.PS       OS.ID
  LOADUC                 IP.PD       IP.PS
  LOADUM                 IP.MECS     IP.PD       IP.PS
  LOADZ                  IP.IACES    IP.MECS     IP.PD      IP.PS       OS.ID
  UCLOAD                 IP.MECS
****819 SUPPORT****

********
  IPARAMS
********
  IP.BCFAP               HLOG        HACT

                         IPARAMS SORTED BY ROUTINE

  HACT                   IP.8CFAP
  HLOG                   IP.8CFAP
****COMPASS****

********
  IPARAMS
********
  IP.PD                  COMPASS     "CP.NAME"
  IP.PS                  COMPASS     "CP.NAME"
  MODEL                  COMPASS     "CP.NAME"
  OS.NAME                COMPASS
  OS.ID                  "CP.NAME"
  OS.VER                 COMPASS
********
  CIOCOM
********
********
  ECSCOM
********
```

IPARAMS SORTED BY ROUTINE

"CP.NAME"	IP.PD	IP.PS	MODEL	OS.ID	
COMPASS	IP.PD	IP.PS	MODEL	OS.NAME	OS.VER

****BASIC ACCESS METHODS****

****
IPARAMS
****

IP.CMU	TXTCRM		
OS.NAME	IOTEXT	SYSTEXT	TXTCRM

IPARAMS SORTED BY ROUTINE

IOTEXT	OS.NAME	
SYSTEXT	OS.NAME	
TXTCRM	IP.CMU	OS.NAME

****ADVANCED ACCESS METHODS****

****
IPARAMS
****

IP.CSET	DICO$AA
OS.NAME	CRA1$AA

IPARAMS SORTED BY ROUTINE

CRA1$AA	OS.NAME
DICO$AA	IP.CSET

****BIT 8****

****
IPARAMS
****

IP.C63	COPY8P			
IP.CSET	T8.HXTB	BDPTAB	T8.6TAB	COPY8P

```
****CE DIAGS****

********
  IPARAMS
********
  IP.YMD                   NORMS
********
  CIOCOM
********

                           IPARAMS SORTED BY ROUTINE

  NORMS                    IP.YMD

****SYMPL****

********
  IPARAMS
********
  OS.ID                    INIT14       INIT40
********
  CIOCOM
********
********
  ECSCOM
********

                           IPARAMS SORTED BY ROUTINE

  INIT14                   OS.ID
  INIT40                   OS.ID

****FTN COMPILER****

********
  IPARAMS
********
  IP.PD                    TSTEXT       FTNMAC       FTNTEXT
  IP.PS                    TSTEXT       FTNMAC       FTNTEXT
  MODEL                    TSTEXT       FTNMAC       FTNTEXT
  OS.NAME                  TSTEXT       FTNMAC       FTNTEXT
  OS.VER                   TSTEXT       FTNMAC       FTNTEXT

                           IPARAMS SORTED BY ROUTINE

  FTNMAC                   IP.PD        IP.PS        MODEL      OS.NAME     OS.VER
  FTNTEXT                  IP.PD        IP.PS        MODEL      OS.NAME     OS.VER
  TSTEXT                   IP.PD        IP.PS        MODEL      OS.NAME     OS.VER

****FTN LIBRARY****
```

****
IPARAMS
****

MODEL	Q2NTRY=

IPARAMS SORTED BY ROUTINE

Q2NTRY=	MODEL

****INTERCOM 5****

****
IPARAMS
****

IP.CMU	STRMOV	KOMSTR				
IP.CPLM	1CI					
IP.C63	ECAFILL					
IP.CSET	3TT	1CI	1IM	1ID	1DS	MES
	MAC	PAGEDAT	ECAFILL	RESEQ		
IP.IACES	1QP					
IP.ILCMD	1CI	1QP				
IP.INT5	T76	3TT	1CI	1QP	1IM	IPP
	LCD	1NI	1DI	1ND	1I1	1NP
	1BR	1QM	1ID	1DS	TBL	FNT
	IUP	IAP	MES	MUJ	MAC	FAD
	DISBEG	STORBEG	FETBEG	REQACT	SCREEN	GETID
	IUID	INTRST				
IP.IUSID	1I1	SETIPS				
IP.MPRIT	1CI	1NP				
IP.IPRLS	1CI	1NP				
IP.X780	T76	3TT	1CI	1QP	1IM	IPP
	LCD	1NI	1DI	1ND	1I1	1NP
	1BR	1QM	1ID	1DS	TBL	FNT
	IUP	IAP	MES	MUJ	MAC	FAD
	DISBEG	STORBEG	FETBEG	REQACT	SCREEN	GETID
	IUID	INTRST				
OS.NAME	RESEQ					

****
ECSCOM
****

IPARAMS SORTED BY ROUTINE

DISBEG	IP.INT5	IP.X780	
ECAFILL	IP.C63	IP.CSET	
FAD	IP.INT5	IP.X780	
FETBEG	IP.INT5	IP.X780	
FNT	IP.INT5	IP.X780	
GETID	IP.INT5	IP.X780	
IAP	IP.INT5	IP.X780	
INTRST	IP.INT5	IP.X780	
IPP	IP.INT5	IP.X780	
IUID	IP.INT5	IP.X780	
IUP	IP.INT5	IP.X780	
KOMSTR	IP.CMU		
LCD	IP.INT5	IP.X780	
MAC	IP.CSET	IP.INT5	IP.X780
MES	IP.CSET	IP.INT5	IP.X780
MUJ	IP.INT5	IP.X780	
PAGEDAT	IP.CSET		
REQACT	IP.INT5	IP.X780	
RESEQ	IP.CSET	OS.NAME	
SCREEN	IP.INT5	IP.X780	
SETIPS	IP.IUSID		
STORBEG	IP.INT5	IP.X780	
STRMOV	IP.CMU		
TBL	IP.INT5	IP.X780	

T76	IP.INT5	IP.X780				
1BR	IP.INT5	IP.X780				
1CI	IP.CPLM	IP.CSET	IP.ILCMD	IP.INT5	IP.MPRIT	IP.IPRLS
	IP.X780					
1DI	IP.INT5	IP.X780				
1DS	IP.CSET	IP.INT5	IP.X780			
1ID	IP.CSET	IP.INT5	IP.X780			
1IM	IP.CSET	IP.INT5	IP.X780			
1II	IP.INT5	IP.IUSID	IP.X780			
1ND	IP.INT5	IP.X780				
1NI	IP.INT5	IP.X780				
1NP	IP.INT5	IP.MPRIT	IP.IPRLS	IP.X780		
1QM	IP.INT5	IP.X780				
1QP	IP.IACES	IP.ILCMD	IP.INT5	IP.X780		
3TT	IP.CSET	IP.INT5	IP.X780			

****BASIC 3****

****
IPARAMS
****

IP.CSET	BASRTS	BASTRNG	BASPRUS	BASCOMP		
IP.PD	BASCARD	BASEGEN				
IP.PS	BASCARD					
OS.NAME	BASOGEN	BASEGEN	BASRTS	BASERRS	BASSINT	BASIGEN
	BASIINP	BASTRNG	BASPRUS	BASCHAN	BASCOMP	BASOPTS

IPARAMS SORTED BY ROUTINE

BASCARD	IP.PD	IP.PS
BASCHAN	OS.NAME	
BASCOMP	IP.CSET	OS.NAME
BASEGEN	IP.PD	OS.NAME
BASERRS	OS.NAME	
BASIGEN	OS.NAME	
BASIINP	OS.NAME	
BASOGEN	OS.NAME	
BASOPTS	OS.NAME	
BASPRUS	IP.CSET	OS.NAME
BASRTS	IP.CSET	OS.NAME
BASSINT	OS.NAME	
BASTRNG	IP.CSET	OS.NAME

****DBU****

```
********
  IPARAMS
********
  OS.NAME                   DF$EXT      DF$EXT$     DF$LCAL
********
  CIOCOM
********
********
  ECSCOM
********

                            IPARAMS SORTED BY ROUTINE

  DF$EXT                    OS.NAME
  DF$EXT$                   OS.NAME
  DF$LCAL                   OS.NAME

****COBOL 5****

********
  IPARAMS
********
  IP.CMU                    CB5TEXT
  IP.YMD                    C$ACCDT
  MODEL                     CB5TEXT
********
  CIOCOM
********
********
  ECSCOM
********

                            IPARAMS SORTED BY ROUTINE

  C$ACCDT                   IP.YMD
  CB5TEXT                   IP.CMU      MODEL
**** FORTRAN 5 ****

********
  IPARAMS
********
  IP.CSET                   LEX
  IP.MFL                    FTN
  IP.PD                     FTN5TXT     FTN         PUC         INIT00      INIT20      FTN510
                            FTN520
  IP.PS                     FTN5TXT     INIT00
  MODEL                     FTN5TXT     FTN         PUC         FEC         FSNAP       LEX
                            HEADER      KEY         DATA        DECL        TYPE        PAR
                            CONRED      LABEL       QCGC        FUN         REG         GEN
                            REC         FAS         MAP         LIST        CCGC        BRIDGE
  OS.NAME                   FTN5TXT
  OS.ID                     FAS
  OS.VER                    FTN5TXT

                            IPARAMS SORTED BY ROUTINE

  BRIDGE                    MODEL
  CCGC                      MODEL
```

CONRED	MODEL					
DATA	MODEL					
DECL	MODEL					
FAS	MODEL	OS.ID				
FEC	MODEL					
FSNAP	MODEL					
FTN	IP.MFL	IP.PD	MODEL			
FTN5TXT	IP.PD	IP.PS	MODEL	OS.NAME	OS.VER	
FTN510	IP.PD					
FTN520	IP.PD					
FUN	MODEL					
GEN	MODEL					
HEADER	MODEL					
INIT00	IP.PD	IP.PS				
INIT20	IP.PD					
KEY	MODEL					
LABEL	MODEL					
LEX	IP.CSET	MODEL				
LIST	MODEL					
MAP	MODEL					
PAR	MODEL					
PUC	IP.PD	MODEL				
QCGC	MODEL					
REC	MODEL					
REG	MODEL					
TYPE	MODEL					

**** FTN 5 LIBRARY ****

****
IPARAMS
****

IP.CSET	Q5NTRY=	COMIO=	ENCODE=	INCOM=	KRAKER=	LDIN=
	OUTC=	OUTF=	COLSEQ=	FERCAP=		
MODEL	Q5NTRY=	DECODE=	ENCODE=	FORSYS=	IDAC=	IIFC=
	OIFC=	MOVLEV	READEC	WRITEC	FERCAP=	RPVCAP=
	MATHTEX	XTOY.				
OS.NAME	Q5NTRY=	CONDIS=	FORSYS=	LDOUT=	NAMOUT=	OUTF=
	PAUSE=	RPVCAP=	MATHTEX			
OS.VER	Q5NTRY=	FORSYS=	MATHTEX			

IPARAMS SORTED BY ROUTINE

COLSEQ=	IP.CSET		
COMIO=	IP.CSET		
CONDIS=	OS.NAME		
DECODE=	MODEL		
ENCODE=	IP.CSET	MODEL	
FERCAP=	IP.CSET	MODEL	
FORSYS=	MODEL	OS.NAME	OS.VER
IDAC=	MODEL		
IIFC=	MODEL		
INCOM=	IP.CSET		
KRAKER=	IP.CSET		
LDIN=	IP.CSET		
LDOUT=	OS.NAME		
MATHTEX	MODEL	OS.NAME	OS.VER
MOVLEV	MODEL		
NAMOUT=	OS.NAME		

OIFC=	MODEL			
OUTC=	IP.CSET			
OUTF=	IP.CSET	OS.NAME		
PAUSE=	OS.NAME			
Q5NTRY=	IP.CSET	MODEL	OS.NAME	OS.VER
READEC	MODEL			
RPVCAP=	MODEL	OS.NAME		
WRITEC	MODEL			
XTOY.	MODEL			

**** FORTRAN DATA BASE FACILITY ****

****
IPARAMS
****

IP.PD	ARG
OS.NAME	CBIO
OS.ID	PRFX

IPARAMS SORTED BY ROUTINE

ARG	IP.PD
CBIO	OS.NAME
PRFX	OS.ID

**** DATA CATALOGUE ****

****
IPARAMS
****

MODEL	FICATION

IPARAMS SORTED BY ROUTINE

FICATION	MODEL

****CDCS 2****

****
IPARAMS
****

OS.NAME	DBQRFI	DB$ATCM	DB$ATCR	DB$CPT	DB$DMGI	DB$EXT
	DB$IO					

IPARAMS SORTED BY ROUTINE

DB$ATCM	OS.NAME
DB$ATCR	OS.NAME
DB$CPT	OS.NAME
DB$DMGI	OS.NAME
DB$EXT	OS.NAME
DB$IO	OS.NAME
DBQRFI	OS.NAME

****CYBER INTERACTIVE DEBUG****

****
IPARAMS
****

IP.CSET	ASCORD					
MODEL	INTERP					
OS.NAME	DBGRPV	DBUG.	DBUGOM	GETPOS	GETSTAT	MAXLDR
	MINIGO	MININI	MINSYS	RTNALL	SETRPV	SPSBRT
	SWAPMI	SWAPUI	SWAPUO	WEBUG	DEBUG	
OS.ID	MAXLDR					

IPARAMS SORTED BY ROUTINE

ASCORD	IP.CSET
DBGRPV	OS.NAME
DBUG.	OS.NAME

DBUGOM	OS.NAME	
DEBUG	OS.NAME	
GETPOS	OS.NAME	
GETSTAT	OS.NAME	
INTERP	MODEL	
MAXLDR	OS.NAME	OS.ID
MINIGO	OS.NAME	
MININI	OS.NAME	
MINSYS	OS.NAME	
RTNALL	OS.NAME	
SETRPV	OS.NAME	
SPSBRT	OS.NAME	
SWAPMI	OS.NAME	
SWAPUI	OS.NAME	
SWAPUO	OS.NAME	
WEBUG	OS.NAME	
RTSTEXT	IP.C63	

****COMMON CODE GENERATOR****

****
IPARAMS
****

MODEL	GPO
OS.ID	CGIA

IPARAMS SORTED BY ROUTINE

CGIA	OS.ID
GPO	MODEL

GLOSSARY

A

AAM

Advanced Access Methods.

AIP

Application Interface Program.

AK

Actual key (a CRM file organization).

ANS/ANSI

American National Standards Institute.

APF

Attached permanent file.

APL

A Programming Language.

APR

Automatic Program Sequencer.

ATS

Advanced Tape Subsystem.

BAM

Basic Access Methods (sequential and word addressable).

BASIC

Beginner's All-Purpose Symbolic Instruction Code.

BCC

Batched corrective code (a release with bug fixes only).

BCD

Binary Coded Decimal.

BINEDIT

On-line binary patch utility enabling users to patch selected records of CTI or HIVS.

BLP

Boolean List Processor (used to process MIP alternate keys).

BNP

Off-line binary patch utility.

BOI

Beginning of Information.

bps

Bits per second.

C45

COBOL 4 to 5 Conversion Aid.

CCI

Communications Control program for INTERCOM.

CCL

CYBER Control Language.

CDCS

CYBER Database Control System.

CE

Customer engineer.

CEJ

Central exchange jump.

CID

CYBER Interactive Debug.

CIO

Circular input/output.

CIP

CYBER Initialization Package.

CLA

Communications line adapter (hardware).

CM

Central memory.

CML

Concurrent Maintenance Library.

CMM

Common Memory Manager.

CMR

Central memory resident (operating system executive).

CMU

Compare and move unit

COBOL

Common Business Oriented Language.

Coldstart

Procedure used to deadstart if the tape or disk controller has not yet been loaded with controlware or the controlware is not running.

COMM

Communications.

Common Deck

A deck that is written on a compile file as a result of a CALL directive.

COMPASS

Comprehensive Assembler.

CPMTR

Central processor monitor program.

CPU

Central Processing Unit.

CR

Carriage return.

CRM

CYBER Record Manager.

CTI

Common Testing and Initialization.

CYBER Initialization Package

A single release tape that provides CTI, HIVS/MSL, EI, and microcode to simplify installation and distribution of these modules.

DA

Direct Access (a CRM file organization).

DABA

Dynamic area base address (used in CMM).

DAT

Device access table.

DBU

Data base utilities.

DDL

Data Description Language (used by CDCS and QU to describe the data base).

DDP

Distributive data path.

DDT

Dismountable device table.

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.

Debugging Facility

Capability within COBOL 5 compiler and execution routines that implements the DEBUGGING MODE clause, lines with D in column 7, and debugging declarative sections.

Debugging Line

Any line with D in column 7 of the source program. Compiled as executable code if DEBUGGING MODE clause is used or DB=DL parameter appears on compiler call; otherwise compiled as comment line.

Deck List

A list internal to Update that contains the names of all decks in the program library and the location of the first word for each deck.

Device Set

A group of rotating mass storage devices. No device can belong to more than one device set. Every file must be contained within one device set, but can be on different devices in that device set.

DML

Data Manipulation Language.

DOCK

FORTRAN extended 4 source language utility for extracting listable internal maintenance specification information from a COMPILE file generated from the NOS/BE program library.

DS

Deadstart (initialization of the operating system).

DSD

Dynamic system display.

DSDI

Deadstart dump interpreter.

DST

Device status table.

EBCDIC

Extended Binary Coded Decimal Interchange Code.

ECS

Extended core storage.

EDITLIB

A NOS/BE utility program for creating and maintaining a library of programs suitable for loading and execution.

EDITOR Mode

The state of INTERCOM existing after the EDITOR utility has been called. A file can be created or modified in the edit file work area. Contrast with command mode.

EEC

Enable error correction.

EOF

End of file.

EOI

End of information.

EOP

End of partition.

EOR

End of record.

EOS

End of section.

ESM

Extended semiconductor memory. NOS/BE uses this type of extended memory in ECS mode only.

EST

Equipment status table.

Exchange Jump

Execution of a CPU program is initiated by an exchange jump. The program is defined by the contents of the exchange package area before the exchange jump took place. For the program to execute, the proper contents of its operational registers must be loaded into the CPU. These contents are what is contained in the exchange package area associated with the program.

EXPORT

The Executive Processor of Remote Tasks which resides in each peripheral processor at the central site used for remote communications. It transmits batch jobs to the central site and controls communication between the remote terminals and NOS/BE.

Extended Memory

An extension to central memory. Types of extended memory are Extended Core Storage (ECS), Extended Semiconductor Memory (ESM), Large Central Memory (LCM), Large Central Memory Extended (LCME), and Unified Extended Memory (UEM).

F45

FORTRAN 4 to 5 Conversion Aid.

FCL

FORTRAN Common Library.

FCO

Field Change Order. The directive to install changes to equipment after the normal manufacturing process in order that the equipment will perform to its written or implied specification.

FDB

File definition block.

FDBF

FORTRAN Data Base Facility.

FET

File environment table (used for OS/user communications).

Field Length

The area in central memory allocated to a particular job; the only part of central memory that a job can directly access.

FIT

File information table (used for CRM/user communications).

FL

Field length.

FMD

Fixed module disk (885).

FNT

File name table.

FOL

Fast overlay loader.

FORM

File Organization and Record Manager.

FORTRAN

Formula Translation language.

FST

File status table.

FTN

FORTRAN.

FWA

First word address.

GCR

Group coded recording.

GEMINI

A control point program providing input and output file load leveling between two linked mainframes under the control of NOS/BE.

H/W

Hardware.

Hardware Initialization and Verification Software (HIVS)

The software package that assists CTI during deadstart and provides deadstart confidence-level testing (HVS). Refer to CYBER Initialization Package.

Hardware Verification Sequence (HVS)

HVS is a subset of HIVS. It tests the ability of memory to hold patterns of data and execute instructions. You can choose to test central memory, extended memory, PP memory, and central processor memory (refer to the V option on the *O* display).

HHA

Highest high address (used with CMM).

HID

Host ID.

HSBT

High speed batch terminal.

ID

Identifiers. This can refer to port/subport, nodes, lines, links, or terminals. Any hardware element or connection can have an ID, normally a sequentially assigned number.

IDT

Logical identifier table.

IEC

Inhibit error correction.

IS

Indexed sequential (a CRM file organization).

JCA

Job control area.

JDT

Job descriptor table.

Job

A set of control statements and the data and directives used by those statements. It begins with a job statement and ends with an EOI statement. Refer to Job Deck.

Job Deck

The physical representation of a job, before execution, as a deck of cards or a group of card images. The first section of the deck begins with a job statement and contains control statements which are used to control the job. Following sections contain the programs and data which the job requires for execution of the control statements. The job deck is terminated by a 6/7/8/9 card. Cards with 7/8/9 multipunched in column 1 separate sections within the deck.

Job Name

The name of an input file assigned by the operating system. Job name is equivalent to file name when speaking of input files or punch/print files carrying the name of the job that generated them.

LCC

Local Communication Controller (for the HSBT, MSBT, LSBT).

LCM

Large central memory.

LCME

Large central memory extended.

LCN

Loosely coupled network.

LFT

Logical flaw table.

LID

Logical ID.

LIP

Link Interface Package/Program.

Loader

A software product that prepares programs for execution by placing program instructions and data blocks in central memory and linking references in the program to the appropriate external routines.

LSBT

Low speed batch terminal.

LWA

Last word address.

MB

Megabytes/bits.

MEJ

Monitor exchange jump (refer to CEJ).

MFL

Minimum field length.

MHF

Maintenance Host Facility.

MIP

Multiple index processor (applies only to IS, DA, AK).

MLIA

Multiplex loop interface adapter.

MMF

Multimainframe (usually means two).

MSBT

Medium speed batch terminal.

MSL

Maintenance Software Library.

MST

Mounted set table.

MTR

Monitor (PP monitor program).

MUJ

Multiple user job (INTERCOM capability and class of job).

Multiplexer

A 6673 or 6674 Data Set Controller which provides hardware interface between the computer data channel and the data set adapter servicing a particular data set.

NAD

Network access device.

NEWPL

New program library.

NOS/BE

Network Operating System/Batch Environment.

NPU

Network processing unit.

NSD

Number of hash points.

OLDPL

Old program library (same as PL)

Overlay

A portion of a program, consisting of one or more modules, which can share an allocated area of memory with others of its kind. When access to a particular module is required, the overlay containing that module is loaded, thus overlaying the previous contents of the memory area allocated for that overlay.

Programs organized into overlays execute in an overlay environment. Such a scheme allows large programs to execute in a limited amount of memory.

PDC

Programmable device controller.

Peripheral Device

An I/O device attached to the NPU A/Q channel. The NPU console is a peripheral device.

Permanent File

A file stored on mass storage. This file is cataloged by the system so that its location and identification are always known to the system. Permanent files cannot be destroyed accidentally during normal system operation. They are protected by the system from unauthorized access according to privacy controls specified when they are created.

PF

Permanent file.

PFC

Permanent file catalog.

PFD

Permanent file directory.

PFT

Physical flaw table.

PID

Physical link ID.

PL

Program library (refers to source tapes).

PP

Peripheral processor.

PPSD

Pages per subdirectory.

PPU

Peripheral Processing Unit, e.g., first level peripheral processor used only to access 819 disks.

PRU

Physical record unit.

QU

Query update.

RB

Record block.

RBR

Record block reservation.

RDF

Remote Diagnostic Facility.

RFL

Request field length.

RHF

Remote Host Facility.

RMS

Rotating Mass Storage.

S/M

Sort/Merge 5.

S/W

Software.

SCB

System circular buffers.

SCED

MUJ capability for COBOL.

SCM

Small central memory.

SCOPE 2

Operating System for 7600 and model 76.

SCOPE 3.x

Batch-oriented operating system, predecessor to NOS/BE.

SCP

System control point.

SPM

Stack Processor Manager.

SPOTS

Spun-off tasks.

SPRM

System Programmer's Reference Manual.

SPY

PPU performance tool used to measure CPU activity.

SQ

Sequential (a CRM file organization).

SSM

Small semiconductor memory.

SYMPL

Symbolic Programming Language.

System Device

An RMS unit that holds the system library and other system files. A system device must be a member of the system set, but not all members of the system set must be system devices.

System Library

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.

TCU

Trunk control unit.

TPM

Two-port multiplexer.

TS

Time-sharing.

UCP

User control point.

UEM

Unified extended memory.

VSN

Volume serial number.

WA

Word addressable (a CRM file organization).

Warmstart

Procedure used to deadstart if the tape or disk controller is loaded and the controlware is running.

INDEX

AAM (Advanced Access Methods) II-4-1
AAM Installation parameters II-4-2
Abnormal termination I-1-19
ACCFILE procedure II-44-5
ACCOUNT I-1-19
ACCTVAL micro II-1-96
ADD statement II-28-31,32
Advanced access methods (AAM) II-4-1
AFL.INT II-1-30
AFL.RMF II-1-30
Allocation style II-28-7
Application accounting definition II-42-8
Application definitions II-42-20
ASYNC macro II-28-25
AUDITKEY II-1-85
AUTO macro II-27-2
AUTOMAT procedure I-1-14
Automatic assignment bit II-1-21
Automatic scratch status II-1-21

BAM II-3-1
BASE I-1-16; II-1-51,52,53
Basic access methods (BAM) II-3-1
BASIC 3 II-21-1
BCC tape I-1-2
BINEDIT (PL1K) I-1-9,25; II-1-1
BISYNC macro II-28-28
Build techniques I-1-5

CATALOG I-1-18
CCI II-29-1
    Installation parameters II-29-5
CCL II-26-1
CDCS 2 I-1-17,18; II-30-1
CE diagnostics II-7-1
CEJ/MEJ switch II-1-4
Central memory II-1-75; II-44-3
Change orders II-1-4
Character sets I-1-17
CHARGE statement II-42-22
CHAR64 II-1-85
CHA statement II-28-31,33
CIA II-8-2
CID II-32-1
CIOCOM II-1-18
CIOCP I-1-18
CIP (CYBER Initialization Package) II-1-2,3,5,54,87
CMM II-1-2,3,85
CMR I-1-17,20,22; II-1-3,52,90
    Configuration parameters II-1-26
    Equipment configuration examples II-1-38,49
    Multiplexer table II-28-9
CMR multiplexer table II-40-2
CMR table structures II-1-32
CMRLIB I-1-6
CMRS I-1-20
CMRSIZE II-1-52
CMR.PD II-1-26
CMR.PS II-1-26
CMR.PW II-1-26
CMS test I-1-18
COBOL 4 to COBOL 5 conversion I-1-18; II-25-1
COBOL 5 II-24-1
Coldstart deadstart II-1-3
Coldstart deck re-creation II-1-87
Command table structure (1CI) II-28-29
Common common decks II-2-1
Common decks I-1-2,19,23,24
Common LIDs II-1-103
Common memory manager II-1-85
Common test and initialization
    Binary release II-1-3
    Program library number I-1-25
Communications Control INTERCOM II-29-1
COMPASS II-2-1
COMPCOM I-1-22
Computing physical addresses II-1-47
COMP3$ II-2-1
COMTBL deck II-28-29
COM2CC macro II-28-10
Configuration, equipment II-1-33
Controller type II-1-35
Controlware binary creation II-1-87
Controlware dump files, accounting II-1-106
Conversion aids
    COBOL 4 to COBOL 5 I-1-18; II-25-1
    FORTRAN 4 to FORTRAN 5 II-37-1
    Maintenance package II-8-2
COPY8P II-5-1
CP.OVLIB II-2-1
CPCTEXT I-1-24
CPMET II-8-2
CPMTR II-1-77
CREATE II-4-1

CRM (refer to CYBER record manager)
Cross-reference lists III-1-1
CTI (refer to Common test and initialization)
CT3 II-1-4
CWARE I-1-16,20
CYBER control language II-26-1
CYBER cross system II-15-1
CYBER database control system 2 II-30-1
CYBER Initialization Package (refer to CIP)
CYBER interactive debug II-32-1
CYBER LOADER II-16-1
CYBER record manager I-1-23,25; II-3-1,4-1

Data base utilities II-22-1
Data Catalog 2 II-41-1
Data description language 3 II-31-1
DBU II-22-1
DDL 3 II-31-1
Deadstart
    Disk deadstart device preparation
      II-1-54
    Dump analyzer II-1-106
    Installation Parameters (DSLCOM)
      II-1-51
    Loading the operating system II-1-86
DEBUG I-1-17
DEBUG code II-42-7
DEBUG statement II-28-29
Deck options I-1-16,18
DECLKEY II-1-85
DEL statement II-28-31,34
Dependencies, product I-1-8
Device set installation II-1-63
Device type codes II-1-37
DIM I-1-18
Disk space threshold II-1-66
DMGMNT I-1-18; II-19-1
DOCK II-8-8,9
DOLLAR I-1-4
Downline load modules II-29-1
DPFC macro II-1-24
DRIVBFL II-1-52
DRIVBUF II-1-52
Driver type selection II-7-1
DSLCOM II-1-51,52
DSPLCHAN II-1-52
DSPLCTLR II-1-52
DSTDEF macro II-1-67
DST1 I-1-5,20,21; II-1-90
DST2 I-1-5,20,22; II-1-90
DST3 I-1-7,22; II-1-90
Dual-state preparation II-44-1
DYNAMFL II-1-85
D44 tests II-17-1

ECS I-1-17
    Deadstart initialization II-1-51
    Definition II-1-52
    Link II-1-37,84
    Segment activity count II-1-82
    System segments II-1-82
ECSBUF macro II-1-79
ECSCOM II-1-78,93
ECSLABEL macro II-1-80
ECSPART macro II-1-81
EDITKEY II-1-85
EDITLIB I-1-6; II-1-93
EDITOR
    Array dimensions in IPFTN
      II-28-21
    DEBUG code II-28-21
    Dependency chart II-28-20
    Installation parameters
      II-28-12,16
Empty common decks I-1-20
EMPTY macro II-28-25
Enhanced station support I-1-18
Equipment configuration II-1-33
Equipment status table (refer to EST)
EST (Equipment Status Table)
  II-1-32,33,14-1,28-9,40-2
ES4IMS I-1-18
ES4XREF I-1-18
EXIT I-1-19
EXPCOM macro II-28-10
EXPORT HS II-40-1
Extended memory (also refer to ECS)
    Console entries II-1-76
    Installation parameters II-1-78
    Label II-1-80
    Residency II-1-93
EXTOVLP II-1-85

Factory format support (844) II-17-1
FCL II-34-1
FCOs II-1-4
FDBF II-35-1
FLAW table II-1-46
FORM II-6-1
FORMAT/FDP II-17-1
FORMAT utility II-17-3
    Console messages II-17-5
    Dayfile messages II-17-5
FORTRAN
    Common library II-34-1,39-1
    Data Base Facility II-35-1
    FORTRAN Extended 4 I-1-17,24; II-9-1
    FORTRAN 4 to FORTRAN 5 conversion aid
      II-37-1
    FORTRAN 5 II-38-1

FTFS Installation II-42-8
FTNMAC I-1-25; II-9-2,3
FTNTEXT II-9-2,3

Gemini installation II-1-103,104
GENTBL macro II-4-3

Hardware feature micro II-1-8
HD I-1-16
HELLO7 I-1-18
HF.LIST micro II-1-8
HIVS (HIVSE) I-1-9,10; II-1-2,4
HY I-1-16

ID table II-1-30
Installation
- Decks I-1-1
- General parameters (IPARAMS) II-1-8
- Job extraction I-1-14
- Order I-1-9
- Parameters
  - ACNMAXC II-42-6
  - FETBUFSIZE II-42-7
  - MAXACN II-43-2
  - MAXFILEXFR II-42-7
  - MAXRTRY II-42-6
  - MAXTCN II-43-2
  - MUJFETS II-43-1
  - MUJTBL$L II-43-2
  - MUJUSRS II-43-2
  - NACOUNT II-43-2
  - TBLLENG II-43-2
  - THRSHLD II-43-2
  - TIMEOUT II-42-6
  - TRMINLN/TRMOTLN II-43-2
- Parameters (IP.symbols)
  - IP.ACNT II-1-10
  - IP.ARCH II-1-11
  - IP.AS II-21-2
  - IP.ATT II-26-3
  - IP.BL II-21-2
  - IP.BSIM II-28-7
  - IP.BUFFE II-28-8
  - IP.CBKSP II-1-18
  - IP.CERNT II-1-11
  - IP.CMPID II-1-81
  - IP.CMU II-1-10
  - IP.CP II-1-11
  - IP.CPLM II-1-11
  - IP.CR II-1-12
  - IP.CSET II-1-8,12,41,21-1
  - IP.CYSTP II-1-84
  - IP.C176 II-1-12
  - IP.DBAL II-1-11
  - IP.DCT II-1-28
  - IP.DPF II-26-2
  - IP.DPFN II-26-2
  - IP.DSRMS II-1-51
  - IP.EBUF II-1-78
  - IP.ECLNK II-1-84
  - IP.ECSB II-1-12,84
  - IP.ECSLK II-1-84
  - IP.ECSTP II-1-27
  - IP.ECSW II-1-29,79
  - IP.ECYC II-1-84
  - IP.EIDLE II-1-84
  - IP.ELIB II-1-78
  - IP.ELST II-1-27
  - IP.ENAP II-1-12
  - IP.ERES II-1-79
  - IP.EXP II-26-2
  - IP.FLINC II-16-2
  - IP.FLMSG II-16-2
  - IP.FP II-26-2
  - IP.FPC II-26-1
  - IP.FTHRL II-1-28
  - IP.FTHRU II-1-28
  - IP.FTNTS II-28-4
  - IP.IACES II-1-12,28-4,8
  - IP.ID II-26-3,28-5
  - IP.IDFL II-28-5
  - IP.IDFLE II-28-6
  - IP.IHEAD II-28-6
  - IP.IHSBS II-28-7
  - IP.IHSLW II-28-7
  - IP.IIBMN II-28-6
  - IP.IIBMX II-28-6
  - IP.ILCMD II-28-8
  - IP.IM3BS II-28-6
  - IP.IM3LW II-28-6
  - IP.IND II-28-7
  - IP.IOLM II-1-13
  - IP.IPRLS II-28-6
  - IP.IQD II-1-13
  - IP.IQPW II-1-13
  - IP.ISFL II-28-7
  - IP.IUSID II-28-9
  - IP.I27BS II-28-6
  - IP.I27LW II-28-6
  - IP.I37BS II-28-6
  - IP.I37LW II-28-7
  - IP.I4ABS II-28-6
  - IP.I4ALW II-28-6
  - IP.I4CBS II-28-6
  - IP.I4CLW II-28-6
  - IP.LCS II-26-2
  - IP.LDBG II-16-2
  - IP.LDER II-16-2
  - IP.LINK II-1-8,13,102
  - IP.LNKBF II-1-84

IP.LNKMN II-1-84
IP.LRT II-16-2
IP.LVF II-1-13
IP.MALOC II-28-7
IP.MAP II-16-3
IP.MAXBL II-1-84
IP.MCPU II-1-13
IP.MECS II-1-14
IP.MFL II-1-28
IP.MMS II-1-14
IP.MPPU II-1-14
IP.MPR II-1-14
IP.MPRIT II-28-7
IP.MSCT II-1-14
IP.MSLM II-1-14
IP.MTL II-1-14
IP.MXCOR II-28-7
IP.NBCD II-1-18
IP.NBRK II-1-18
IP.NDEN II-1-6,14
IP.NDFS II-1-14
IP.NJFL II-1-14
IP.NOISE II-1-19
IP.NOIS9 II-1-19
IP.NPV II-26-2
IP.NTCN II-1-19
IP.OPRI II-1-15
IP.OQD II-1-15
IP.PD II-1-15
IP.PFRP II-1-15,65
IP.PNL II-26-2
IP.POSFL II-1-15
IP.PRIX II-28-8
IP.PRUB II-28-8
IP.PS II-1-15
IP.PSET II-16-2
IP.PTCN II-1-19
IP.PW II-1-15
IP.RBINT II-1-66,67
IP.RCYC II-1-19
IP.REW II-16-2
IP.RLD II-26-3
IP.RM II-1-15
IP.RPE1 II-1-19
IP.RPE2 II-1-19
IP.SCHDE II-1-6,16
IP.SCL II-26-3
IP.SCS II-26-1
IP.SECS II-1-16
IP.SEQ II-1-16
IP.SFL II-1-16
IP.SIDLE II-1-16,77
IP.SIOL II-1-16
IP.SLIB II-1-29,90
IP.SMS II-1-16
IP.SPR II-1-16
IP.SPT II-1-16
IP.SRMS I-1-18; II-1-17
IP.STEX II-1-102
IP.STL II-1-17
IP.SYS II-26-3
IP.SYSE II-1-29
IP.SYSL1 II-1-28
IP.TAPO II-26-2
IP.TCPUB II-1-17,31
IP.TDEN II-1-17
IP.TSG II-1-7,19
IP.TSL II-28-8
IP.TYPE II-1-17
IP.UP II-1-17
IP.US II-1-17
IP.VER II-1-29,26-3
IP.WEC II-1-23
IP.XJ II-1-5
IP.X780 II-28-8
IP.YMD II-1-18
IP.7LNK II-1-102
IP.819 II-1-18
Requirements I-1-10
Sequence I-1-9
User name for NOS/VE II-44-2
Installation procedures
AAM II-4-5
CDC MODEL 171 configuration II-27-1
NOS/BE II-1-7
Integer multiply II-1-4
Interactive Transfer Facility (ITF) II-43-1
INTERCOM output file priority II-28-7
INTERCOM 5 II-27-5,II-28-1
IOTEXT I-1-23
IPARAMS II-1-8
IPCOM II-28-12
IPFILL II-28-12
IPFTN II-28-12
IPTEXT I-1-22; II-7-2
IRADR II-1-51
IRP (RMS driver) II-1-52
ITF (Interactive Transfer Facility) II-43-1
ITF verification II-43-3
IXGEN II-4-1

JANUS,spacing code arrays II-1-24
Job Classes II-1-69

KYAN II-4-1

LABELMS II-1-44
LE.DFBXX II-1-27
LE.DFB00 II-1-27
LIBS I-1-21
Line speeds II-28-26
LIST I-1-16
Load leveling II-1-100,102,103,105

LOADER II-16-1
Local NAD definitions II-42-15
LOCALCC I-1-19,20
LOCALIN I-1-19
LOCLIB I-1-18
Logical ID table II-1-30
Loosely coupled network II-42-16,43-1
L.APF II-1-29,65
L.ECSSWP II-1-5,29
L.ELST II-1-27
L.EST II-1-27
L.FNT II-1-27
L.IDT II-1-27
L.INS II-1-27
L.LDIS II-1-27
L.SCHJCA II-1-30
L.SCHJDT II-1-30
L.SEQ II-1-27

Maintenance package II-8-1
Maintenance tools II-8-1
Master device II-1-35
Maximum NDR definition II-42-12
MAXRBCNT II-1-51
MFLINK delay file II-42-10
MFSTAT II-1-94
MHF (Maintenance Host Facility) II-42-9
MINI I-1-21
MIPGEN II-4-1
Miscellaneous (automatic assignment) bits II-1-22
MODEL micro II-1-18
MODE4 macro II-28-25
MODLEVEL micro II-21-1
MORENUC I-1-20
MOREOV I-1-20
MOREPP I-1-20
MORESYS I-1-20
MTS I-1-16; II-1-35
MUJ
- IPARAMS II-28-21
- Macro II-28-10
- System errors II-28-23
- Table II-28-11

MUJTBL macro II-28-11
Multiplexer definition entries II-28-25
MUXCR macro II-27-2
MUX2550 macro II-28-25

NAD II-42-2,14
NAD initialization II-42-5
NADIBM I-1-17
NADMIN I-1-17
Network access device (refer to NAD)
Network configuration statements II-42-13
NOCCP II-15-3
Noise bracketing II-1-18
NOISEL II-7-2
Nomenclature, installation deck I-1-13
NOS/VE dual-state II-44-1
NO1M1 I-1-17
NPU
- Configuration aids II-29-8
- Programs II-28-23

NVEBUILD deck II-44-1
N.BRKPT II-1-28
N.CP II-1-28
N.DEVICE II-1-28
N.RBR II-1-28
N.RQS II-1-28
N.SBSYS II-1-29; II-44-2
N.SETS II-1-29,65
N.SPRPP II-1-28
N.VDDT II-1-29
N.VRNBUF II-1-28

OLMS II-7-1
On-line maintenance software II-7-1
Operator Option Matrix II-1-51
OPTDF micro II-1-51
OPTFTN I-1-17
Option dependencies II-1-23
OPTION macro II-1-25
Order of installation I-1-9
OS.ID micro II-1-18
OSMINIT II-29-2
Overcommitment bit II-1-22
OVL macro II-1-82

Parity retries II-1-19
Partner jobs II-44-5
Pascal I-1-18
Password
- System dayfile II-1-24
- Universal II-1-65
- Utility II-28-30

PE I-1-16
Permanent file installation II-1-63
PFC 580 printer arrays II-1-24
PL naming conventions I-1-13
PL1AT I-1-23
PMD II-34-1,39-1
PMODKEY II-1-85
Polling times II-1-97
Port definition entries II-28-25
Post mortem dumps (PMD) II-34-1,39-1
PPTEXT II-1-5
Prestaging bits II-1-21
Programmable format control (PFC) 580 printer arrays II-1-24
PR.IDLE II-1-77
PR.BATCH II-1-77
PR.INT II-1-77

PR.SCP II-1-77
PR.SYS II-1-77

Q.ILNOFC II-28-8
QLOG, accounting II-1-94
QTF initiation II-42-8
QUERY UPDATE II-19-1

RBTSIZE II-1-52
RCFGEN II-42-11
RDF (Remote Diagnostic Facility) II-1-2,14-1
READS I-1-19
REASON I-1-2,16
Record block reservation table II-1-41
Remote Diagnostic Facility (refer to RDF)
Remote Host Facility II-42-1
REMOTE macro II-28-11
Remote Mainframe
- Definition II-42-13
- LID definitions II-42-13
- Path definitions II-42-14

Retention cycle II-1-19
RHF II-42-1
RHF configuration file generation II-42-11
RHF support II-1-93
RMS
- Deadstart II-1-17
- Device sets II-1-63
- Driver (IRP) II-1-52

RUNIRHF procedure II-44-5

SAVE I-1-18
SAVEPL I-1-19
SCED II-28-34
Scheduling parameters II-1-69,71
SDFRD micro II-1-24
SDFXR micro II-1-24
Segment activity count II-1-82
SIFT II-8-2
SIP II-8-4
SIS63 II-8-2,9
Sort/Merge 5 I-1-27; II-11-1
Spacing code arrays (580 PFC printer) II-1-24
SPOT jobs II-1-94
SPOTs, accounting II-1-96
SPY II-8-2
SRMS I-1-18
SRMSTXT I-1-25
Station buffer sizes II-1-99
Station support II-1-94
Symmetric/replacement station II-1-96
SYMPL I-1-10; II-8-1
SYNGEN II-19-1,31-1
SYSPROC I-1-20
System
- Circular buffers (SCB) II-1-94
- Dayfile password II-1-24
- Residency II-1-92
- Seconds II-1-31
- Texts I-1-23,24,25

S.AUTO II-1-19,21
S.DEBUG II-1-20,22
S.NOOR II-1-20,22
S.OCJI II-1-7,20,22,23
S.PREA II-1-20,22
S.PRES II-1-20,21
S.PSON II-1-20,22
S.SCBL II-1-20
S.SCEL II-1-20
S.SCUL II-1-20
S.TRDO II-1-20,23
S.TSEC II-1-20,23
S.UEOJ II-1-20,23
S.URES II-1-20,21
S.2LBP II-1-20,22

Tape creation, deadstart I-1-6
- Density II-1-19
- Processing parameters II-1-18
- Scheduling notes II-1-6,7
- Scheduling parameters II-1-19

Tape scheduling options II-1-19
TBL assembly options II-28-11
TBUFO II-1-51
Template for partner jobs II-44-5
Threshold, disk space II-1-66
TSFTN I-1-17
TSTEXT II-9-2,3

ULIB I-1-18,21
Update installation options II-1-85

VALIDAC control statement II-1-96
Validating users II-44-7
Verification jobs I-1-13

Weighting constants, system seconds II-1-31

XTRALIBS I-1-20

0MT I-1-16; II-1-87
0SJ I-1-16; II-1-87
0SY I-1-16; II-1-87
0SZ I-1-16; II-1-87
1QP II-28-11
2CS II-28-11
63CSET I-1-17
66x tape subsystem I-1-16
8-bit subroutines II-5-1
819 RMS II-1-5
844 RMS II-1-38

COMMENT SHEET

MANUAL TITLE: CDC NOS/BE Version 1 Installation Handbook

PUBLICATION NO.: 60494300 **REVISION:** AB

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

CUT ALONG LINE

REV. 5/86 PRINTED IN U.S.A.

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

CORPORATE HEADQUARTERS, P.O. BOX 0, MINNEAPOLIS, MINN. 55440
SALES OFFICES AND SERVICE CENTERS IN MAJOR CITIES THROUGHOUT THE WORLD

LITHO IN U.S.A.

CONTROL DATA CORPORATION