

OS/390 Batch and Utilities

(North Pre-Titan and South System)

April 2002



National Institutes of Health
Center for Information Technology
NIH Computer Center
12 South Drive MSC 5607
Bethesda, Maryland 20892-5607

Publication No. CIT174

About this Manual

The NIH Computer Center is migrating to a single OS/390 system named Titan. This new system will ultimately host all of the applications that previously ran on the OS/390 North System and the OS/390 South System. At this time, the OS/390 North System transition to Titan is complete and the OS/390 South System transition is underway.

CIT is preparing documentation describing the new batch processing techniques that are native on Titan. However, since the South System is still in use we have republished this manual describing batch processing and utilities. Additionally, since most South and North JCL (job control language) will continue to work on Titan, we anticipate that this manual will continue to have value for those applications that are not yet using native Titan JCL.

The OS/390 South System information is current. Therefore, South users who submit batch jobs should use this manual as a reference.

The North System information in this manual consists of "frozen" documentation that describes the system before the Titan transition. **Please note: Most former North System JCL for submitting batch jobs should work on Titan.** However, some North System software, referred to in the "Utilities" section, is no longer available.

1	INTRODUCTION.....	1
2	SOUTH – DEFINING AND CONTROLLING A BATCH JOB.....	3
2.1	JOB CONTROL LANGUAGE OVERVIEW	3
2.1.1	Operating System Overview	3
2.1.2	Job Control Language Explained.....	4
2.2	STANDARDS FOR BATCH JOBS.....	4
2.2.1	Maximum Limits for Batch Jobs	4
2.2.2	Restrictions at NIH	6
2.2.3	Job Class Summary.....	7
2.2.3.1	Job Turnaround	8
2.2.4	Job Class Standards.....	9
2.2.4.1	Job Class A	9
2.2.4.2	Job Class E.....	9
2.2.4.3	Job Class B.....	10
2.2.4.4	Job Class C.....	10
2.2.4.5	Job Class F	11
2.2.4.6	Job Class H	11
2.3	JOB STATEMENT.....	11
2.3.1	Simplified JOB Statement.....	12
2.3.2	Complete JOB Statement.....	13
2.3.3	Accounting Information Parameter.....	14
2.3.4	COND Parameter	15
2.3.5	MSGLEVEL Parameter	16
2.3.6	REGION Parameter	16
2.3.7	RESTART Parameter.....	17
2.3.8	USER Parameter	18
2.4	EXEC (EXECUTE) STATEMENT	18
2.4.1	PGM Parameter.....	20
2.4.2	PROC and Procedure Name Parameters.....	21
2.4.3	COND Parameter	21
2.4.4	PARM Parameter	23
2.4.5	REGION Parameter	23
2.5	DD (DATA DEFINITION) STATEMENT.....	24
2.5.1	Input Stream Data Set Parameter (*,DATA,DLM)	26
2.5.2	Output Stream Data Set Parameters (SYSOUT,COPIES,DEST).....	28
2.5.3	DCB (Data Control Block) Parameter	30
2.5.4	DCB Subparameters.....	31
2.5.4.1	BLKSIZE, LRECL and RECFM	31
2.5.4.2	DSORG, BUFNO and OPTCD	34
2.5.5	Additional DCB Subparameters for Tapes	36
2.5.6	DDNAME Parameter.....	37
2.5.7	DISP Parameter.....	37
2.5.8	DSNAME Parameter	40

2.5.9	Dummy Parameter	42
2.5.10	LABEL Parameter	43
2.5.11	SPACE Parameter	44
2.5.12	UNIT DD Parameter	46
2.5.13	VOLUME Parameter	49
2.5.14	AVGREC Parameter	52
2.5.15	LIKE Parameter	53
2.5.16	DATACLAS Parameter	54
2.5.17	Concatenating Data Sets	61
2.6	CREATING AND USING DATA SETS	61
2.6.1	Creating Temporary Data Sets on System Scratch Disks	61
2.6.2	Creating and Deleting Data Sets	62
2.6.3	Creating Permanent or Temporary Data Sets on Tape	63
2.6.4	Using Permanent Data Sets	64
2.6.5	Using a Passed Data Set	65
2.7	CATALOGED PROCEDURES	66
2.7.1	JCLLIB	66
2.7.2	Defining Symbolic Parameters in a Cataloged Procedure	67
2.7.3	Using Cataloged Procedures	67
2.7.4	Assigning Values to and Nullifying Symbolic Parameters	68
2.7.5	Modifying Parameters on the EXEC Statement	68
2.7.6	Overriding or Adding DD Statements	69
2.7.7	Instream Procedures	70
2.7.8	INCLUDE Groups	71
2.8	JOB DEPENDENCY CONTROL STATEMENTS	72
2.8.1	/*AFTER and /*BEFORE	73
2.8.2	/*CNTL	73
2.9	RESOURCE CONTROL STATEMENTS	75
2.9.1	/*MESSAGE	75
2.9.2	/*ROUTE XEQ	76
2.9.3	/*ASSIGN	77
2.9.4	/*ACCESS	77
2.10	JOB EXECUTION CONTROL STATEMENTS	78
2.10.1	/*KEYWORD	78
2.10.2	/*NOTIFY	78
2.10.3	/*DISCOUNT	78
2.10.4	/*QUICK	79
2.10.5	/*PRIORITY	79
2.10.6	/*RERUN	80
2.10.7	/*PASSWORD	80
2.10.8	/*XMIT	80
2.10.9	/*NETACCT	81
2.10.10	/*ROUTE XEQ LATE	81
2.11	OUTPUT CONTROL STATEMENTS	81
2.11.1	/*ROUTE	81
2.11.2	/*JOBOUT	82

2.11.3	/*DDOUT	84
2.11.4	OUTPUT JCL Statement	86
2.11.5	/*NOPURGE.....	88
2.11.6	Combining Printing Statements	88
2.12	PRINTING CONTROL STATEMENTS	90
2.12.1	/*TITLE	90
2.12.2	WYLBUR Printing Control Statements.....	90
2.13	COMMENT STATEMENT	91
2.14	AVOIDING JOB TURNAROUND DELAYS.....	91
2.15	INTERACTIVE FACILITIES FOR BATCH JOB SUBMISSION	92
2.15.1	Batch Job Submission through the Web	92
2.15.2	Automatic Scheduled Batch Job Submission through WYLBUR.....	93
2.16	UNDERSTANDING JOB OUTPUT	94
3	SOUTH – SERVICE AND UTILITY PROGRAMS.....	99
3.1	UTILITY PUBLICATIONS.....	100
3.2	UTILITY RESTRICTIONS.....	100
3.3	MANIPULATING CATALOGED DATA SETS	101
3.3.1	CATDS	102
3.3.2	UNCATDS.....	103
3.3.3	VDSUTIL	103
3.4	GENERAL DATA SET UTILITIES.....	104
3.4.1	Summary of General Data Set Utilities	105
3.4.2	Copying Data Sets.....	105
3.4.2.1	COPY	105
3.4.2.2	DSCOPY.....	107
3.4.3	Printing Data Sets	109
3.4.3.1	Summary of Print Utilities	110
3.4.3.2	PRINT	110
3.4.3.3	DSLISIT	113
3.4.3.4	EDSLIST.....	114
3.4.3.5	POSTSCRIP for PostScript Printing.....	116
3.4.4	Sorting and Merging Data Sets	117
3.4.4.1	SORT	119
3.4.4.2	EDSSORT.....	120
3.4.4.3	MERGE.....	122
3.4.5	COMPARE for Comparing Data Sets	123
3.4.6	Scrambling and Unscrambling Data Sets	124
3.4.6.1	DSSCM and DSUNSCM.....	124
3.4.6.2	DSSCMI and DSUNSCMI	125
3.4.7	Transmitting Data to or from Tapes at Remotes.....	126
3.4.7.1	RMTDSIN.....	126
3.4.7.2	RMTDSOUT.....	128
3.4.8	ADSMAP for Obtaining Data Set Information	130
3.4.9	ADSERASE to Erase Sensitive Data.....	133
3.4.10	EDSUTIL for Data Set Manipulation.....	135
3.5	TAPE AND TAPE DATA SET UTILITIES.....	135

3.5.1	Summary of Tape Utilities.....	136
3.5.2	TAPEMAP for Dumping the Labels of a Tape	139
3.5.3	PALTAPE for Processing Problem Tapes.....	141
3.5.4	TAPECOPY for Copying Multi-Data-Set Tapes.....	142
3.5.5	RESPOUT for Reprocessing an Offline Output Tape	143
3.5.6	VOLSTAT for Setting and Showing Volume Status.....	144
3.5.7	TAPESCAN.....	149
3.6	DISK DATA SET UTILITIES	149
3.6.1	Summary of Disk Utilities	150
3.6.2	Saving and Retrieving Disk Data Sets on Tape	152
3.6.2.1	DSSAVE	156
3.6.2.2	DSGET	157
3.6.2.3	EDSSAVE.....	159
3.6.2.4	EDSGET	161
3.6.2.5	PDSSAVE.....	163
3.6.2.6	PDSGET	164
3.6.2.7	DDSSAVE	166
3.6.2.8	DDSGET.....	167
3.6.2.9	Hints on Saving and Retrieving Data Sets.....	169
3.6.3	DSRENAME for Renaming Disk Data Sets.....	170
3.6.4	Scratching Disk Data Sets.....	171
3.6.4.1	DSSCR for Scratching Data Sets.....	171
3.6.4.2	DSSCRV for Scratching Non-cataloged Data Sets	172
3.6.5	Creating and Maintaining Partitioned Data Sets.....	172
3.6.5.1	PDSALLOC	174
3.6.5.2	PDSDEL	175
3.6.5.3	MEMADD	176
3.6.5.4	MEMRENAM.....	177
3.6.5.5	MEMALTER	178
3.6.5.6	PDSMAP.....	180
3.6.5.7	USRPDSL	180
3.6.5.8	PDSPACK.....	181
3.6.5.9	PDSCOPY.....	182
3.6.5.10	USRPDSC.....	184
3.6.5.11	Other PDS Utilities	185
3.6.6	Printing Disk Data Sets.....	186
3.6.6.1	PDSLIST.....	186
3.6.6.2	DDSLIST	189
3.6.7	DDSCOPY for Copying a Direct Data Set.....	190
3.7	MAINTAINING DEDICATED DISK VOLUMES.....	191
3.7.1	Guidelines for Disk Maintenance	191
3.7.1.1	Technical Coordinator	191
3.7.1.2	Backups.....	191
3.7.1.3	DISKMAP for Checking for Missing or Overlapping Tracks.....	192
3.7.1.4	DISKMAP Status Codes.....	192
3.7.1.5	Emergency Procedures.....	192

3.7.1.6	Compressing a Disk	193
3.7.1.7	Hints on Maintaining Dedicated Disks	194
3.7.2	DISKMAP for Mapping Private Disks	194
3.7.3	Backing Up Dedicated Disks	198
3.7.3.1	DISKSAVE	199
3.7.3.2	DISKEXPT	200
3.7.4	ADSRECOV for Recovering Data Sets from Dedicated Volumes	201
3.7.5	DISKGET for Restoring Dedicated Disks	213
3.8	PROCESSING EDIT FORMAT DATA SETS WITH BATCH JOBS	214
3.8.1	Summary of Utilities for EDIT Format Data Sets	215
3.8.2	Reading and Writing EDIT Format Data Sets	217
3.8.2.1	EDSIN	217
3.8.2.2	EDSOUT	219
3.9	USING THE BINDER	222
3.9.1	How to Estimate Region Size	223
3.9.2	Specifying Load Module Blocksize	224
3.10	USING THE LOADER	225
3.11	DSSUBMIT A PROGRAM TO SUBMIT BATCH JOBS	226
3.12	JCLSCAN TO CHECK FOR JCL ERRORS	227
3.13	PASBILL FOR LISTING LAST MONTH'S COMPUTER BILL	228
3.14	GENERATION DATA SETS	228
3.14.1.1	The Model Data Set Label	229
3.14.2	The Cataloged Index	230
3.14.3	Creating a New GDG Data Set	230
3.14.4	Deleting a Generation Data Group	231
3.15	EXECUTING WYLBUR COMMANDS AS A BATCH JOB	231
3.16	HOST-TO-HOST FILE TRANSFER	234
3.16.1	SENDFILE and RCVFILE	234
3.16.2	CONNECT:Direct	238
4	SOUTH – MESSAGES AND CODES	241
4.1	ABENDS AND CRITICAL MESSAGES	241
4.1.1	Common ABENDs	241
4.1.2	Critical Messages	243
4.2	GENERAL MESSAGES (NIH, HASP, IEF, MIM)	244
4.3	DISK AND MSS MESSAGES	247
4.4	TAPE HANDLING MESSAGES	250
4.5	JOB STATEMENT ERROR MESSAGES	255
4.6	WHERE TO FIND OTHER MESSAGES	255
5	NORTH – DEFINING AND CONTROLLING A BATCH JOB	257
5.1	JOB CONTROL LANGUAGE OVERVIEW	257
5.2	JOB STATEMENT	257
5.2.1	Job Class Standards	261
5.3	JCLLIB STATEMENT	263
5.4	EXEC STATEMENT	263
5.5	DD (DATA DEFINITION) STATEMENT	265

5.5.1	Data Definition Name Field (ddname)	266
5.5.2	DD Parameters for Creating Data sets	269
5.5.3	Disk Storage and Management	279
5.5.3.1	Scratch Packs	279
5.5.3.2	Public Storage User Packs	280
5.5.3.3	Disk Backup	280
5.5.3.4	Space Management of Disk Data sets	280
5.6	JOB ENTRY SUBSYSTEM VERSION 2 (JES2)	284
5.6.1	JES2 Job Control Statements	284
5.6.2	JES2 RJE Control Statements	291
5.7	JOB SUBMISSION	292
5.7.1	Remote Job Entry (RJE)	292
5.7.2	Interactive Job Submission	292
5.7.3	Web Job Submission	292
5.7.4	Batch Job Cancellation	292
6	NORTH - SERVICE AND UTILITY PROGRAMS AND PROCEDURES	293
6.1	IBM UTILITIES	293
6.2	ACCESS METHOD SERVICES	294
6.3	MAX	294
6.4	SYNCSORT OS/VS UTILITY (SORT/MERGE/COPY)	296
6.5	ONLINE SYNCSORT (OLS)	298
6.6	FDRDSF (FAST DUMP/RESTORE)	298
6.7	PCCVTOC	301
6.8	HOST-TO-HOST FILE TRANSFER	301
6.8.1	Network Job Entry (NJE)	301
6.8.2	File Transfer Facility (SENDFILE, RCVFILE)	304
6.8.3	CONNECT:Direct (Network Data Mover)	308
6.9	TAPE INFORMATION AND UTILITIES	310
6.9.1	Tape Management, Security, and Storage	310
6.9.1.1	Tape Management System TMS/CA1 Features	310
6.9.1.2	TMS Error Conditions	312
6.9.1.3	Tape Security System (TSS)	312
6.9.1.4	Data Encryption, Subroutine Method	315
6.9.1.5	Tape Standards	316
6.9.1.6	Tape Inventory Management	317
6.9.1.7	Procedure for Purchasing Tape	319
6.9.1.8	Foreign Tape Processing	320
6.10	SANITIZE FOR ERASING DATA	324
6.11	VISION:UILDER (MARK IV) CATALOGED PROCEDURES	326
7	NORTH - MESSAGES AND CODES	329
7.1	COMMON ABENDS	329
7.2	TMS COMPLETION CODES AND MESSAGES	332
7.2.1	TMS SYSTEM ABEND CODES	332
7.3	JES2 MESSAGE LOG - TMS MESSAGES AND REPLIES	335
7.4	OS/390 NORTH SYSTEM RACF MESSAGES	336

7.5	ABR ONLINE MESSAGES FOR DATA SET RETRIEVAL	341
8	APPENDIX.....	343
8.1	APPENDIX – SELECTED TITAN TRANSITION WEB PAGES	344
8.2	APPENDIX – SELECTED TITAN /SOUTH SYSTEM NEWS ARTICLES	379
8.3	APPENDIX – ADDITIONAL NJE INFORMATION FOR TITAN	382
9	INDEX.....	383

1 INTRODUCTION

This manual is for users who run batch jobs on the OS/390 systems at the NIH Computer Center, Center for Information Technology (CIT). It describes the job control language (JCL) statements required for running batch jobs on the OS/390 South System and the former North System as well as utility programs available for these two systems.

When South System customers and applications migrate to Titan, there will be a single OS/390 system operating at the NIH Computer Center. This manual presents current South System information and the former North System information that was "frozen" as of July 31, 2001 when the North System migrated to Titan. Current Titan users who are comfortable with the North System JCL and procedures can continue to use this manual as a reference. A new Titan batch processing manual will be published in the near future.

Be sure to look at the Appendix of this manual for information relating to the transition to Titan.

Note: Most existing North and South JCL (job control language) for submitting batch jobs will work on Titan with minimal changes. Some North facilities may no longer work. For information on changes related to the transition, go to:

<http://silk.nih.gov/silk/titan>

Additional information on the NIH Computer Center is available at:

<http://datacenter.cit.nih.gov>

The information in this manual should be used in conjunction with the *NIH Computer Center User's Guide* (South System), the *Titan User's Guide*, other CIT publications, and related IBM manuals. Titan and South users should refer to the *IBM OS/390 MVS JCL User's Guide*, GC28-1758 and the *IBM OS/390 MVS JCL Reference*, GC28-1757. The current manuals for each system can be ordered from CIT (see below).

Changes that affect job control language, batch processing, and utility programs will be fully tested and pre-announced through:

- *Interface* (<http://datacenter.cit.nih.gov/interface>), the technical newsletter that covers the services and facilities provided by the IT to NIH and other government agencies
- "Titan/South System News," an online mail facility. Join the NIH Listserv list CIT-Titan-News at <http://list.nih.gov/archives/cit-titan-news.html>.

The South System batch processing facilities and utilities described in this manual receive full support from the Center for Information Technology. Questions should be directed to the CIT Technical Assistance and Support Section (TASC).

Readers of this manual are assumed to have an understanding of elementary JCL. In all format statements given in this manual, key words are shown in upper case and must be coded exactly as they appear in the format statements. Words or letters in lower case represent variables for which specific information must be substituted when the parameters are coded.

Assistance

CIT offers several ways of requesting assistance:

- telephone:
Call 301-59 **GO-CIT** (301-594-6248)
- e-mail:
Send e-mail to tasc@nih.gov
- Web-based customer support:
<http://support.cit.nih.gov>
- Web-based Service Request Ticket (SRT) system:
<http://datacenter.cit.nih.gov/srt>

Publication Services

- The CIT publication ordering service allows users to order hardcopy publications, view publications online, or print publications at the central printers or on their workstation printers. Users can order, print, or view many CIT publications and manuals relating to the OS/390 operating system and job control language. This service is available at:

<http://publications.cit.nih.gov>

Contact TASC if you require assistance.

- Titan/North users can access IBM BookManager to view online technical documentation. From the CIT/Titan Primary Option panel, select C for Products, then B for books.

2 SOUTH – DEFINING AND CONTROLLING A BATCH JOB

This section includes the job control language (JCL) statements required to describe the resources needed by a batch job running at this installation. Readers of this section are assumed to have an understanding of elementary JCL. Additional details for any JCL statements may be obtained from the *IBM OS/390 MVS JCL User's Guide*, GC28-1758 and the *IBM OS/390 MVS JCL Reference*, GC28-1757.

Note: A modified version of WYLBUR will eventually be placed on Titan. CIT encourages users to consider alternative long-term strategies through the WYLBUR Transition Working Group. Refer to *Interface* for information related to WYLBUR.

To keep informed about WYLBUR transition issues and to join the RETIRE-WYL listserv list go to:

<http://silk.nih.gov/silk/retirewyl>

2.1 JOB CONTROL LANGUAGE OVERVIEW

Existing JCL (job control language) for submitting batch jobs should work on Titan with minimal changes. For information on changes that will occur when the South System migrates to Titan, go to:

<http://silk.nih.gov/silk/titan>

This section provides an overview of job control language describing its interaction with the IBM OS/390 Operating System and listing related publications.

2.1.1 Operating System Overview

The OS/390 Operating System introduces programs to the computing system, initiates their execution, and provides them with all the resources and services necessary for them to do their work. It is made up of a general library of programs that can be tailored to meet many requirements. The installation can select the systems programs that it needs, add its own programs to them, and update existing programs, as needs change.

For illustrative purposes, the programs and routines that compose the operating system are classified as a control program and processing programs. The three main functions of the control program are to accept and schedule jobs in a continuous flow (job management); supervise on a priority basis each unit of work to be done (task management); and simplify

retrieval of all data, regardless of the way it is organized and stored (data management). The processing programs consist of language translators (such as the FORTRAN compiler), service programs (such as the Binder), and problem programs (such as users' programs). The processing programs are used to define the work that the computing system is to do and to simplify program preparation. For a description of the operating system currently in use at the NIH Computer Center, see the *NIH Computer Center User's Guide*.

2.1.2 Job Control Language Explained

Job control language (JCL) is the means by which the user communicates with the operating system. Through JCL, the user tells the computer what to do with programs, data sets, and I/O devices. JCL is the only means of getting a batch job on and off the computer; therefore, every user must understand JCL.

JCL is checked in two steps.

- Conversion processing is done when the job first enters the job queue. During conversion, syntax errors are checked, and all JCL (including expanding procedures) is resolved.
- Later, at interpretation, the system detects other errors, including misplaced JCL statements.

If a job fails during conversion due to invalid syntax of the JCL and the job is corrected and resubmitted, "logical" errors might still be found during interpretation. Users will be informed of failures if the jobs in question have the NOTIFY attribute. This is particularly useful for jobs which are run overnight. The syntax errors will be detected, and the job will print immediately or be placed in the OUTPUT HOLD queue giving the user a chance to correct the JCL and resubmit the job. If a job is directed to OUTPUT HOLD, the output from the conversion phase will be available when that phase is completed. When the converter encounters an error, a user signed onto WYLBUR or TSO will be notified if the job has the NOTIFY attribute. See Section 3.12 for information on a facility that thoroughly checks JCL before final job submission.

2.2 STANDARDS FOR BATCH JOBS

This section defines the maximum and class-dependent limits for jobs running at NIH and enumerates those JCL parameters that may not be used.

2.2.1 Maximum Limits for Batch Jobs

The following overview defines maximum limits for batch jobs.

Figure 2-1. Maximum Limits for Batch Jobs

LIMITS FOR A SINGLE JOB	
Number of steps	255
Number of instream procedures	15
Region (below the line)	6144K
Lines output	4 million
Scratch disk space allocated at one time (tracks)	100,000
Contention suspension	30 min.
Lines output at Remote 8	5,000
LIMITS ON NUMBER OF JOBS FROM ONE USER	
Total jobs in the system	150
Jobs AWAITING EXECUTION and EXECUTING	50

In addition to the "below the line" region requested by the REGION parameter (below the 16-megabyte line in virtual storage), each job is automatically assigned at least 96MB (98,304K) of requestable virtual storage above the 16-megabyte line (requiring 31-bit addressing for use).

The limits on numbers of jobs apply to jobs submitted by a single user. These limits are necessary to prevent overloading the operating system queues and to allow timely processing of jobs for all users. The limit on total jobs includes jobs in all categories (AWAITING EXECUTION, EXECUTING, AWAITING PRINT, IN OUTPUT HOLD, etc.).

A maximum of 100,000 tracks (4,747,600,000 bytes) of public 3390 disk scratch space can be allocated to a job at any one time. Space allocated to a temporary data set that is passed among the steps of a job is counted as part of this total until it is deleted. If a job exceeds the maximum, it will be cancelled, and no refund will be allowed.

The following maximum limits for batch jobs have more restrictive standards in most job classes (see Section 2.2.4):

CPU time	20,000 seconds
Region (above the line)	512MB
Tape drives in use at one time	7
Reel tape drives in use at one time	2

2.2.2 Restrictions at NIH

The following job control language parameters should not be used at this facility:

Job statement

ADDRSPC	NOTIFY	RD
CLASS	PASSWORD	TIME
GROUP	PERFORM	TYPRUN
	PRTY	

EXEC statement

ACCT	DYNAMNBR	RD
ADDRSPC	PERFORM	TIME
DPRTY		

DD statement

AMP	FLASH	PROTECT
BURST	FREE	QNAME
CHKPT	HOLD	SUBSYS
DSID	MODIFY	TERM
DYNAM	MSVGP	UCS
FCB	OUTLIM	

The DCB, LABEL, SPACE and UNIT parameters all have some restrictions on their use at NIH. The restrictions are listed in the individual descriptions of these parameters in Section 2.5. See Section 3.2 for a detailed list of NIH restrictions on the use of IBM utilities.

In addition, any JES3 parameters not supported by JES2, all JES3 statements, and the following JES2 statements should not be used:

command statements	SETUP
JOBPARM	SIGNOFF
SIGNON	

2.2.3 Job Class Summary

Figure 2-2. Job Class Summary

	CPU TIME (seconds)[*]	LINES PER JOB (1000s)	REGION^{**}	TYPICAL^{***} PROCESSING TURNAROUND	SERVICE^{****} LEVEL OBJECTIVES
Class A					
Maximum	10	4000	6144K	10 sec.	21 sec.
Default	10	20	4096K		
Class E					
Maximum	20000	4000	6144K	24 sec.	1 min. 40 sec.
Default	150	20	4096K		
Class B					
Maximum	100	4000	6144K	1 min. 14 sec.	3 min. 30 sec.
Default	100	20	4096K		
Class C					
Maximum	20000	4000	6144K	1 min. 33 sec.	6 min. 30 sec.
Default	150	20	4096K		
Class F					
Maximum	9999	4000	6144K	N/A	N/A
Default	200	4000	N/A		
Class H					
Maximum	20000	4000	6144K	N/A	N/A
Default	300	20	4096K	N/A	N/A

Notes:

* CPU time is specified in 3090 J seconds.

** In addition to any requested region "below the line" in the 24-bit address space, all jobs are automatically assigned 96MB in the 31-bit address space.

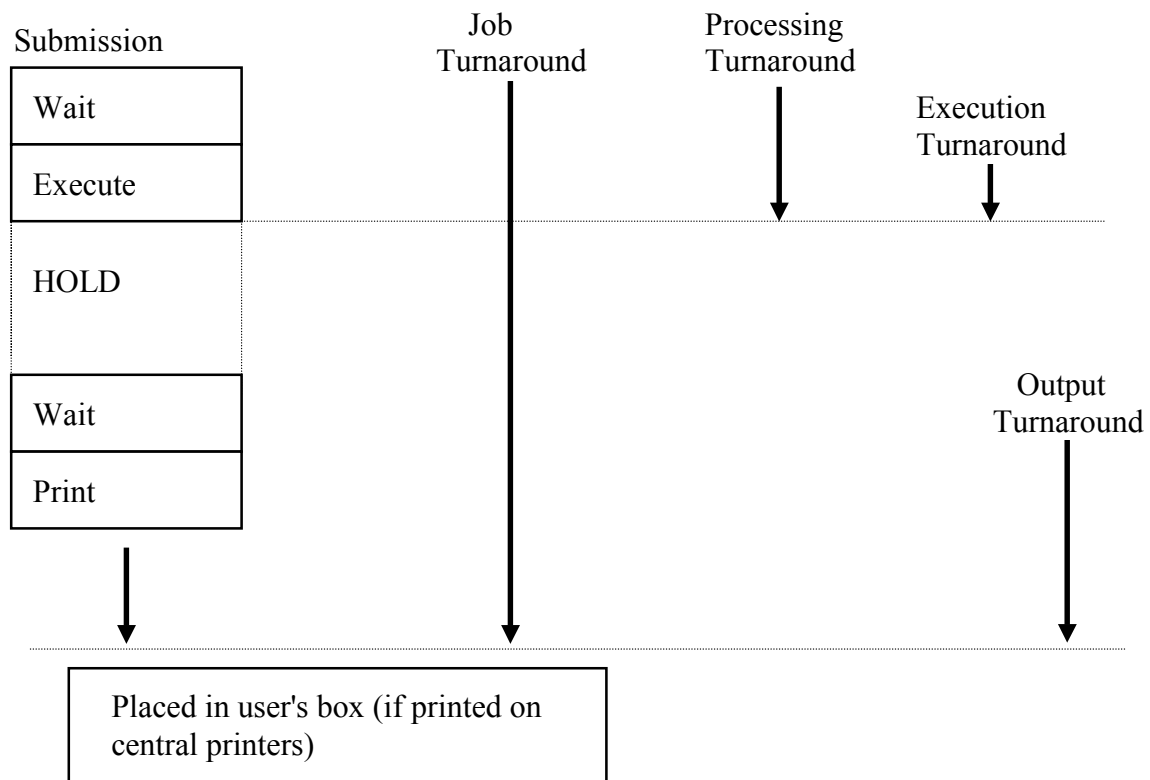
*** "Typical" processing turnaround times are the median turnaround times observed from 10/1/94 through 12/31/94 for jobs that were run (submitted and completed execution) from 9:00 a.m. to 4:00 p.m.

**** The Service Level Objectives are the "not-to-exceed" targets for mean execution times for the job classes during peak business hours. See Section 2.2.3.1 for more information.

2.2.3.1 Job Turnaround

Job turnaround has several phases, which are explained below.

Figure 2-3. Phases of Job Turnaround



Processing turnaround is the time that elapses from when a job is submitted until it has completed execution. Processing turnaround includes execution turnaround, which is measured from the time a job starts executing until it is completed.

The Service Level Objectives, shown in Figure 2-2, are expressed as the average (mean) execution times for jobs that completed execution during peak business hours (between 10:00 a.m. and 11:00 a.m. and between 2:00 a.m. and 3:00 p.m. on business days). The NIH Computer Center uses these objectives as targets for system performance. The NIH Computer Center strives not to exceed these limits.

When a job has completed execution, it may enter OUTPUT HOLD (from which the user can FETCH the output and selectively examine it at a workstation) or it may be placed in a queue to be printed.

Output turnaround is from the time a job enters the print queue until the output has been placed in the user's output box. This time can vary widely depending on the services chosen. For jobs printed at the central facility with the standard printing requirements (see the *NIH Computer Center User's Guide*), the time for the job to be printed, separated from other jobs,

and placed in the user's output box will normally be two hours or less during non-discount periods.

Delivery times also vary, including at times delays for messenger pick-up and redistribution within the user's building.

WYLBUR's LOCATE command can be used to trace the progress of a job (see the *WYLBUR Fundamentals* manual).

2.2.4 Job Class Standards

Batch jobs running at NIH are divided into classes depending on the amount of system resources and operator intervention they require. It is basic to the philosophy of the NIH Computer Center that the class of a job is influenced only by these factors. Occasionally a system problem may cause a job to remain in execution for far longer than the expected turnaround time for its class. If this occurs for a Class A, B, or E job, please contact the CIT Technical Assistance and Support Center (TASC). See Section 2.14 for a discussion of the causes of delayed job turnaround.

Please note that if your job violates NIH Computer Center standards, it may be cancelled at any time, and no refund will be permitted. Also, any time a user adversely affects throughput for other users, the user's job may be cancelled.

2.2.4.1 Job Class A

This class of work provides the user with a facility to aid modular debugging and testing of programs, job control language testing, and other runs of very short duration. Class A was designed to help the user during the initial stages of program preparation. By deferring processing on a job that meets Class A standards, costs may be reduced (see Section 2.10.3). To insure the user of rapid turnaround, Class A service imposes the following restrictions in addition to the limits given in Section 2.2.1:

- Maximum and default CPU times are both 10 seconds.
- No tapes of any kind may be requested. This includes scratch tapes.
- Temporary data sets on scratch disks may be used. Data sets in the FILE, TMP, and MSS management classes as well as data sets on dedicated disks may be used.

2.2.4.2 Job Class E

This class of work permits execution of jobs, which require a moderate amount of CPU time but do not require operator intervention. By deferring processing on a job, which meets Class E standards, costs may be reduced (see Section 2.10.3). The following restrictions apply to Class E in addition to those given in Section 2.2.1:

- Maximum and default CPU times are both 150 seconds.

-
- No tapes may be mounted; this includes scratch tapes.
 - Temporary data sets on scratch disks may be used. Data sets in the FILE, TMP, and MSS management classes as well as data sets on dedicated disks may be used.

Note: Job Class E users should be careful when specifying large CPU limits, since a job that goes into a loop could be very expensive. (See the *NIH Computer Center User's Guide* for the policy on refunds.)

2.2.4.3 Job Class B

This class of work is of a developmental or short processing nature. It is that type of work that is necessary to test and debug a program to make sure that it is free of program and logic errors. Class B work is short in duration, requiring some operator intervention. By deferring processing on a job, which meets Class B standards, costs may be reduced; see Section 2.10.3. To insure minimum turnaround time to all users of Class B work, there are the following restrictions in addition to the limits given in Section 2.2.1:

- Maximum and default CPU times are both 75 seconds.
- The following limits are placed on the use of tapes for Class B jobs:
 - A maximum of two tapes may be requested by specific volume serial number.
 - A maximum of seven tape drives may be used simultaneously, including no more than two-reel tape drives. Remember that only two assigned tapes may be used; any others must be scratch tapes.
 - No more than 90 tapes can be assigned in one job step.
 - Scratch tapes used for offline output (e.g., MICRO105) cannot be accessed by the user after the job that creates them is completed.

2.2.4.4 Job Class C

Class C is mainly for production and long running jobs. Depending on system load, Class C jobs may go into execution during daytime hours. More frequently, however, Class C jobs are completed overnight, (that is, jobs submitted by 10:00 p.m. on weeknights are completed by 7:00 a.m. the following morning). For jobs that are intended to run during the discount period, the `/*DISCOUNT` control statement must be contained in the job (see Section 2.10.3). The following restrictions apply:

- Total CPU time must not exceed 20000 seconds; the default is 100 seconds. Please note that a job that uses the maximum permitted CPU time can cost thousands of dollars; estimate accurately; do not casually specify the maximum.
- All Class C jobs should be step-restartable. If a Class C job aborts and has no step restarts, refund will be made only for the step that aborted. For a very large step, the maximum refund will be for one hour of processing (job step charges/elapsed hours).

-
- The following limits apply to the number of tapes which a C class job may use:
 - A maximum of seven tape drives may be used simultaneously, including no more than two reel-to-reel tape drives.
 - Note that tapes that are RETAINED or that contain "passed" data sets may remain mounted during job steps in which they are not used. They should be counted against the totals for these steps.
 - Scratch tapes used for offline output (e.g., MICRO105) cannot be accessed by the user after the job that creates them is completed.
 - No more than 90 tapes can be assigned in one job step.

Note: Job Class C users should be careful when specifying large CPU limits, since a job that goes into a loop could be very expensive. (See the *NIH Computer Center User's Guide* for the policy on refunds.)

2.2.4.5 Job Class F

This job class may not be specified by the user; it is used by the recovery jobs generated by the ADSRECOV procedure and WYLBUR and TSO listings. See 3.7.4 for further information on ADSRECOV. No turnaround time can be given for class F jobs.

2.2.4.6 Job Class H

This class allows development work during the day, without interfering with production applications (e.g., TSO, WYLBUR, DB2, SILK Web, and all other job classes) and without incurring large charges. Class H jobs run at the discount rate during the day when unused capacity exists. These jobs will run at lower priority than other job classes with no turnaround time guarantees, so as not to degrade performance on production work.

A Class H job receives the discount rate during the day only if the following criteria are met:

- No tapes of any kind may be requested; this includes scratch tapes.
- The minimum CPU time must be at least five minutes (300 seconds).

For detailed information on batch job limits, see Section 2.2.1 and Section 2.2.3.

2.3 JOB STATEMENT

The JOB statement indicates to the operating system the beginning point for a job. Every job must have a standard NIH Computer Center JOB statement.

2.3.1 Simplified JOB Statement

A simple form of the JOB statement is:

```
//jobname JOB (aaaa,box,class),name
```

Basic Coding Information

"jobname"	3 to 8 alphanumeric or national (\$, #, @) characters. Characters 1-3 must be the programmer's set of registered initials. If several jobs are being submitted, characters 4-8 of the jobname should be unique, otherwise the turnaround time for the jobs may be increased.
"aaaa"	the programmer's account number.
"box"	the box number representing the programmer's assigned box in Output Distribution Services (one-to-four digits or M, followed by one-to-three digits).
"class"	determined by the resources needed for the job. Small jobs not requiring tapes get fast turnaround from Class A or Class E. Class B or C is needed to mount a tape. B is the default class. See Section 2.2.4 for details on class limits.
"name"	the name of the programmer responsible for the job and contains 1 to 20 characters. If the name contains any characters other than letters, numbers, hyphens or leading or embedded periods, it must be enclosed in single quotes.

Additional Information

- All of the fields above are required. If the simplest form of the JOB statement is used, the job will be run as Class B. This job class permits the use of magnetic tapes and is generally completed in less than two hours. For detailed information on the limitations to Class B, see Section 2.2.1 and Section 2.2.4.3.
- See the *NIH Computer Center User's Guide* for information on how to obtain a valid set of initials and an account number.

Examples

```
//ABCJOB1 JOB (ACT1,33,E),SMITH.J  
  
//XYZ1 JOB (LMNO,M500,A),'MARY JONES'
```

2.3.2 Complete JOB Statement

The complete form of the JOB statement is:

```
//jobname JOB (accounting-information),name,kwd-parms comments
```

The JOB statement consists of the characters // in columns 1 and 2 and six fields: jobname, JOB, (accounting information), name, kwd-parms, and comments.

Usage at NIH

The following parameters cannot be used:

ADDRSPC	PERFORM
CLASS	PRTY
GROUP	RD
NOTIFY	TIME
PASSWORD	TYPRUN

Basic Coding Information

"jobname"	3 to 8 alphanumeric or national (\$,#,@) characters. The first three characters must be the programmer's registered initials.
"accounting-information"	a positional parameter.
"name"	contains 1-to-20 characters and identifies the programmer responsible for the job. If the "name" contains any characters other than letters, numbers, hyphens, or leading or embedded periods, it must be enclosed in single quotes.

- One or more blanks must be coded following the "jobname," JOB and "kwd-parms" fields.
- The following "kwd-parms" parameters can be used, in any order, after the "name" field:

```
COND  
MSGLEVEL  
REGION  
RESTART  
USER
```

Examples

```
//XYZPRINT JOB (ABC1,101),ZEBRA  
//ABCSTAT JOB (XYZ2,74,C),'AHAB CAMEL',USER=EWE
```

2.3.3 Accounting Information Parameter

The accounting information lists basic information needed to process any job.

```
(aaaa,box,cl,tm,lpj,cds)
```

The parentheses are required.

The format of a simple JOB statement containing all of these items would be:

```
//jobname    JOB (aaaa,box,cl,tm,lpj,cds),name
```

Usage at NIH

There are 6 positional subparameters available: the first two subparameters are required, and commas must be supplied for any subparameters that are omitted if further subparameters are to be specified. The subparameters are:

"aaaa"	the account number must be supplied.
"box"	the box number, representing the user's assigned box in Output Distribution Services (one-to-four digits or M, followed by one-to-three digits for mailing purposes). It is required.
"cl"	the class (one letter; A, B, C, E, and F are the legal classes; B is the default).
"tm"	the maximum CPU time in seconds (an integer of one-to-five digits)
"lpj"	the maximum number of lines, in thousands, per job (one-to-four digits).
"cds"	the maximum number of cards punched, in thousands (one-to-three digits). Cards are generally used only at remote sites for file transfer.

Additional Information

- The account number supplied is checked with the three initials that begin the jobname. If they are not a valid set, the user's job will be terminated.
- The estimate for CPU time must allow for a possible plus or minus variance of 15 percent between runs of the same job.
- See Section 2.2.3 for specific maximums and defaults for "tm" and "lpj."
- The time estimate is made to prevent excessive cost if the program gets into a loop. We strongly advise against indiscriminate use of the maximum value of this parameter.
- The maximum number of lines for a job is equal to the number of lines for one copy of output multiplied by the number of copies requested.

Examples

```
//LMNSRT      JOB (ABC2,150,B),WOLF  
  
//XYZPRINT    JOB (ABC1,101,C,40,50),ZEBRA  
  
//ABCSTAT     JOB (XYZ2,74,A,,50),CAMEL
```

```
//FASTJB      JOB (ZIP1,150,E),DEER
```

2.3.4 COND Parameter

The COND parameter specifies whether a job terminates based on return codes issued by one of its job steps.

```
COND= ((code,operator),...)
```

Basic Coding Information

"code" must be a number from 0 through 4095. This number is compared with the return code issued by each job step.

"operator" describes the type of comparison to be made. It may be any one of the following:

GT	greater than
GE	greater than or equal to
EQ	equal to
LT	less than
LE	less than or equal to
NE	not equal to

- The operator is used to compare the code to the return code from each step. If the comparison is true, the remaining job steps are bypassed, and the job is terminated.

Additional Information

- Up to 8 different return code tests may be made for each job. If any of the tests are satisfied, the job is terminated.
- If only one return code is tested, the outer parentheses are not needed.
- Specific return code values (and their meanings) depend upon the program being executed. The compilers and utility programs provided by the NIH Computer Center generally adhere to the following convention:

0	Execution was successful.
4	A warning or informational message was generated. Although the message should be examined, processing was generally successful.
8	An error was encountered.
12	An error was encountered.
16	An error was encountered.

- If the COND parameter is coded on both the JOB statement and on one or more of the job's EXEC statements, satisfaction of the return code test on the JOB statement terminates the job, even if the return code test requested on the EXEC statement is not

satisfied. Likewise, if the return code test requested on the EXEC statement is satisfied, the step is bypassed even if the test requested on the JOB statement is not satisfied.

Examples

```
//XYZPRINT    JOB  (ABC1,101,C),ZEBRA,COND=(8,LT)
//ABCSTAT     JOB  (XYZ,74),CAMEL,COND=((100,GE),(50,LT))
```

2.3.5 MSGLEVEL Parameter

The MSGLEVEL parameter indicates what system output and messages are to be written as part of the job output.

```
MSGLEVEL=(statements,messages)
```

Usage at NIH

- MSGLEVEL=(1,1) is the default and should not be overridden.
- If any value other than 1 is supplied for "messages," it will be ignored—that is, if MSGLEVEL=(0,0) is requested, MSGLEVEL=(0,1) will be used.

2.3.6 REGION Parameter

The REGION parameter specifies the maximum amount of virtual storage to be allocated for a job "below the line."

```
REGION=nnnnK
REGION=nM
```

The character K (for kilobytes) or M (for megabytes) must be given if the REGION parameter is used. K equals 1024 bytes; M equals 1024K.

Basic Coding Information

- If the JOB statement specifies a REGION parameter, any REGION parameters on EXEC statements will be ignored.
- No REGION parameter on a JOB or EXEC statement should request less than 16K.

Additional Information

- It is generally better to code a region size for each job step on its EXEC statement, than to override REGION on the JOB statement. For information concerning how to determine the region requirements for a job step, see Section 3.9.1.

Examples

```
//ABCSTAT      JOB (XYZ2,74,E),CAMEL,REGION=600K
```

2.3.7 RESTART Parameter

The RESTART parameter indicates where the job should begin execution when the user resubmits it after a system failure.

```
RESTART= *  
          stepname  
          stepname.procstepname
```

Usage at NIH

- Checkpoint restarts are not allowed at NIH.

Basic Coding Information

"*"	indicates that execution is to be restarted at the first job step.
"stepname"	specifies that execution is to be restarted at the named job step.
"stepname.procstepname"	specifies that execution is to be restarted at a cataloged procedure step. "stepname" is the name of the job step that calls the cataloged procedure, and "procstepname" is the name of the procedure step.

- None of the steps preceding the RESTART step are executed.

Additional Information

- The RESTART step and all later steps can have no backward references to the nonexecuted steps.
- If the step to be restarted executes a cataloged procedure, the procedure stepname must be included, even if the restart step is the first or only step in the procedure.
- Long-running jobs should have restart points for every step and at least every half hour in long steps. Program logic must be used to provide for restarts within long steps.

Examples

```
//XYZPRINT      JOB (ABCD1,101,E),ZEBRA,RESTART=PRINT  
  
//ABCSTAT      JOB (XYZ2,74),CAMEL,RESTART=STEP2.GO
```

2.3.8 USER Parameter

The USER parameter indicates the RACFID of the user. This parameter is required if the job is to gain other than universal access to RACF-protected data sets. For further information on RACF security, see the *NIH Computer Center User's Guide*.

```
USER=iii
```

or

```
USER=aaaaiii This will run on the South System or Titan
```

Usage at NIH

- If the job is submitted through WYLBUR and WYLBUR is used to create the JOB statement, the RACFID option of the RUN command is used to create the USER field.

Basic Coding Information

"iii" the set of user initials.

- A /*PASSWORD statement (Section 2.10.7) must be included in the job stream unless the job is submitted through WYLBUR or TSO.

Additional Information

- The RACFID does not have to be the same as the initials under which the job is run.
- For details on levels and conditions of access to RACF-protected data sets, see the *NIH Computer Center User's Guide*.

Examples

```
//ABCSTAT JOB (XYZ1,74),CAMEL,USER=ABC
```

```
//XYZPGM JOB (DEF2,74),WOLF,USER=ABC
```

2.4 EXEC (EXECUTE) STATEMENT

The EXEC statement identifies the program or procedure to be executed. This statement must be the first JCL statement of each step in a job.

```
//stepname EXEC parameters comments
```

The EXEC statement consists of the characters // in columns 1 and 2 and four fields:

```
stepname  
EXEC  
parameters  
comments
```

Each field is separated by one or more blanks.

Usage at NIH

The following parameters are not supported:

ACCT	DYNAMNBR	RD
ADDRSPC	PERFORM	TIME
DPRTY		

Basic Coding Information

- One EXEC statement is needed for each job step; it must precede all DD statements for that step.
- A unique stepname is required for each step if:
- Backwards references are made to the step.
- The step is to be referenced by the RESTART parameter on the JOB statement.
- The step is part of a multi-step procedure and must be specifically referenced. A specific reference to the stepname may be required to add or override DD statements to the step, or to add or override parameters on the EXEC statement within the procedure.
- A stepname consists of 1 to 8 alphanumeric or national (\$, #, @) characters beginning with an alphabetic or national character.
- Two types of parameters can be coded in the operand field of the EXEC statement: positional parameters and keyword parameters.

- One of the following three positional parameters must precede any keyword parameter:

PGM
PROC
procedure-name

- The following optional keyword parameters can be used, in any order, after the positional parameter:

COND
PARM
REGION

Additional Information

- If either PROC or the procedure-name parameter is being used on an EXEC statement, symbolic parameters may also appear in the EXEC statement. See Section 2.7.3 for further information.

Examples

```
//STEP1      EXEC   PGM=IEFBR14,COND=(8,LE)

//GOSTEP     EXEC   CBLACALL,REGION=2000K,PARM=UPDATE
```

2.4.1 PGM Parameter

The PGM parameter names the program to be executed in a job step.

PGM=program-name

Basic Coding Information

"program-name" 1 to 8 alphanumeric or national (\$, #, @) characters beginning with an alphabetic or national character.

- The program identified by "program-name" must be stored in the system library (SYS1.LINKLIB), be stored in a private library defined by a STEPLIB DD or JOBLIB DD statement, or be defined by a backward reference.

Additional Information

- A program to be executed must be a fully resolved load module and exist as a member of a partitioned data set (PDS).
- A backward reference may be made to a DD statement that defines, as a member of a partitioned data set, the program to be executed. This method is usually used when a previous job step has created a temporary partitioned data set to store a program until it is needed. The reference may take either of two forms:

PGM=*.stepname.ddname

PGM=*.stepname.procstepname.ddname

"stepname" the name of the step which contains the DD statement. If the DD statement is within a procedure, "procstepname" must be used to identify the step.

"ddname" the name of the DD statement being referenced.

Examples

```
//LOAD      EXEC   PGM=IEWL
//SYSLMOD   DD   DSN=&&GOSET(MAIN),UNIT=SYSDA,
//           DISP=(,PASS,DELETE),
//           SPACE=(CYL,(2,1,1),RLSE)
//GO        EXEC   PGM=*.LOAD.SYSLMOD,REGION=300K
```

2.4.2 PROC and Procedure Name Parameters

The procedure-name (or PROC=procedure-name) parameter identifies the procedure to be executed. For additional information concerning the use of procedures, see Section 2.7.3.

procedure-name or PROC=procedure-name

The procedure-name may be specified with or without the PROC keyword.

Basic Coding Information

"procedure-name" 1 to 8 alphanumeric or national (\$, #, @) characters beginning with an alphabetic or national character.

- The procedure must contain EXEC statements for the programs to be executed.
- A procedure must be an instream procedure or be stored in a procedure library. (Remember, procedure libraries cannot be stored on the MSS.)

Additional Information

- If an EXEC statement specifies a multi-step procedure, the keyword parameters should generally be qualified to specify the steps to which they apply.

Examples

```
//COST      EXEC  PASBILL
```

```
//STEP#2    EXEC  FORVLKGO , PARM.LOAD=XREF , REGION.GO=200K
```

2.4.3 COND Parameter

The COND parameter specifies whether or not a job step is to be executed, based on return codes (or ABENDs) from previous steps.

COND=(code,operator) or COND=abend-test

Additional capabilities of the COND parameter are described in the IBM manual *OS/390 MVS JCL Reference*, GC28-1757.

Basic Coding Information

"code" must be a number from 0 through 4095. This number is compared with the return codes issued in all previous steps.

"operator" describes the type of comparison to be made. The operator may be any one of the following:

GT	greater than
GE	greater than or equal to

EQ	equal to
LT	less than
LE	less than or equal to
NE	not equal to

The operator is used to compare the code to the return code from each previous step. If the comparison is TRUE, then the current step (with the COND parameter) is bypassed.

"abend-test" may be either EVEN or ONLY.

EVEN specifies that the job step is to be executed even if a previous job step has ABENDED.

ONLY specifies that the job step is to be executed only if a previous job step has ABENDED.

Additional Information

- Specific return code values (and their meanings) depend upon the program being executed. The compilers and utility programs provided by the NIH Computer Center generally adhere to the following convention:

0	Execution was successful.
4	A warning or informational message was generated. Although the message should be examined, processing was generally successful.
8	An error was encountered.
12	An error was encountered.
16	An error was encountered.
- Up to 8 different return code tests may be made for each job step. If any of the tests are satisfied, the step is bypassed.
- If only one return code is tested, the outer parentheses are not needed.
- If the COND parameter is coded on both the JOB statement and on one or more of the job's EXEC statements, satisfaction of the return code test on the JOB statement terminates the job, even if the return code test requested on the EXEC statement is not satisfied. Likewise, if the return code test requested on the EXEC statement is satisfied, the step is bypassed even if the test requested on the JOB statement is not satisfied.

Examples

```
//STEP1      EXEC   DSSAVE ,COND=( 8 , LE )

//MYSTEP      EXEC   PRINT ,COND=ONLY

//RUNSTEP     EXEC   PGM=MYPGM ,COND=( 16 , EQ )
```

2.4.4 PARM Parameter

The PARM parameter specifies variable information to be passed to the program to be executed.

```
PARM=value
```

Basic Coding Information

"value" may contain up to 100 characters of information.

- If the value contains any characters that are not alphanumeric or national (\$, #, @), the entire value must be enclosed in apostrophes (single quotes). If these "special characters" are commas, the value may be enclosed in parentheses instead of apostrophes. The enclosing apostrophes (or parentheses) are not passed to the program being executed.
- If the value contains an ampersand or an apostrophe, two ampersands or apostrophes must be given. The program being executed will receive only one character.

Additional Information

- If the value must be continued on another statement, it should be enclosed in parentheses and continued at a comma. The continuation comma is considered part of the value field and counts toward the maximum of 100 characters.
- Including a keyword as a value will cause it to be printed in the job output. To preserve keyword confidentiality, examine the program documentation for an alternative (e.g., SYSIN) that will allow the keyword to be masked.

Examples

```
//STEPONE     EXEC   PGM=IEWL , PARM=MAP

//S5          EXEC   PLIXCALL , PARM=( JONES , JOHN , FEBRUARY15 ,
//          1945 , BETHESDA , MARYLAND )

//STEP26      EXEC   PGM=MYPGM , PARM=( ' QUADRATIC , AX+BX**2 ' )
```

2.4.5 REGION Parameter

The REGION parameter specifies the maximum amount of virtual storage to be made available to a job step "below the line."

```
REGION=nnnnK
```

REGION=nM

The character K (for kilobytes) or M (for megabytes) must be given if the REGION parameter is used. K equals 1024 bytes; M equals 1024K.

Usage at NIH

- The maximum region that can be requested is 6144K (6M). A default of 4096K is assigned to each job step when REGION is not specified (i.e., in your JCL or in the JCL of the procedure being executed).
- All jobs and TSO sessions are automatically assigned 96 megabytes in the extended 31-bit address space. This region size is not requested on or controlled by JCL.

Basic Coding Information

- If the JOB statement specifies a REGION parameter, REGION parameters on EXEC statements will be ignored. It is better practice to specify the REGION parameter on the EXEC statements than on the JOB statement.
- For information concerning how to determine the region requirements for a job step, see 3.9.1.

Additional Information

- Some programs will use all of the region which is made available to them. Since this can cause an increase in job cost, it may be advantageous to use the REGION parameter to restrict the available region.
- The system allocates region in increments of 4K. Hence, a request for 201K will be interpreted as though 204K were specified.

Examples

```
//RUNSTEP3 EXEC  PGM=MYREPORT,REGION=200K  
//STEP2      EXEC  CBLALKGO,REGION.GO=600K
```

2.5 DD (DATA DEFINITION) STATEMENT

A DD statement describes each input and output data set used by the program defined on the EXEC statement in the job step. The description includes the location and characteristics of the data set or indicates system input or output facilities to be used.

```
//ddname DD parameters comments
```

The DD statement consists of the characters // in columns 1 and 2 and four fields: ddname, DD, parameters, and comments. Each field is separated by one or more blanks.

Usage at NIH

- The following ddnames cannot be used:

JOBCAT	SYSCKEOV
STPCAT	SYSMDUMP
SYSCHK	

- The following parameters are not supported:

AMP	FLASH	PROTECT
BURST	FREE	QNAME
CHKPT	HOLD	SUBSYS
DSID	MODIFY	TERM
DYNAM	MSVGP	UCS
FCB	OUTLIM	

- Special disposition processing is performed for JOBLIB and STEPLIB DD statements (see Section 2.5.7).

Basic Coding Information

"ddname" consists of 1 to 8 alphanumeric or national (\$, #, @) characters beginning with an alphabetic or national character

- All job steps require one or more DD statements to define the input and output requirements of the program being executed.
- The following ddnames imply special functions and should not be used for other purposes:

JOBLIB	SYSABEND
STEPLIB	SYSUDUMP

- The ddname provides the link between a program's files (for program input and output) and the location and characteristics of the data set or the system facilities to be used.
- Each ddname within a job step should be unique.
- If one of the following positional parameters is used, it must precede any keyword parameter:

*

DATA
DUMMY

- The following keyword parameters can be used, in any order; however they must follow a positional parameter if one is present:

AVRGREC	DEST	LIKE
CHARS	DISP	SPACE
COPIES	DLM	SYSOUT
DATACLAS	DSNAME	UNIT
DCB	LABEL	VOLUME
DDNAME		

Additional Information

- If duplicate ddnames exist within a job step, allocation of devices and space, and disposition processing are done for all DD statements; however, all references to the ddname are directed to the first such DD statement in the step.
- The ddname must always be coded except in the following two instances:
 - A DD statement defines a data set that is concatenated with a data set defined by the preceding DD statement.
 - The DD statement is the second or third consecutive DD statement that defines an indexed sequential data set.
- One job step can have no more than 1635 DD statements.

Examples

```
//STEP1    EXEC  PGM=MYREPORT
//OUTPUT   DD    DSNAME=AAAAIII.FINAL,DISP=(NEW,KEEP),
//          VOLUME=(PRIVATE,SER=999999),UNIT=TAPE
//INPUT     DD *
//          .
//          .
//          INPUT DATA
//          .
//          .
//ERRORS    DD  SYSOUT=A

//MULTI     DD    DSNAME=AAAAIII.DATA1,UNIT=TAPE,DISP=SHR,
//          VOL=(PRIVATE,SER=009999)
//          DD    DSNAME=AAAAIII.DATA2,
//          DISP=SHR,UNIT=TAPE,
//          VOLUME=(PRIVATE,SER=123456)
```

2.5.1 Input Stream Data Set Parameter (*,DATA,DLM)

The DD * or DD DATA statement indicates that data immediately follows.

```
//ddname   DD *
//ddname   DD DATA
//ddname   DD *,DLM=aa
```

```
//ddname DD DATA,DLM=aa
```

Basic Coding Information

- If DD * is used, the input data set is terminated by any statement which contains // or /* in columns 1 and 2.
- If DD DATA is used, only /* in columns 1 and 2 will be recognized as a delimiter (// statements will be treated as data).
- The DLM parameter is optional. It may be used to specify an alternate delimiter for the input data set. "aa" indicates the two-character delimiter which must appear in columns 1 and 2 of the first statement following the data. (Both // and /* statements will be treated as data, if DLM is used with DD DATA. /* will be treated as data if DLM is used with DD *, and // will remain as a delimiter).
- All job stream input data must be processed with a logical record length of 80 characters (LRECL=80), and the BLKSIZE must be a multiple of 80.

Additional Information

More than one input stream data set can be used in a job step if the ddnames are unique. Each data set must be followed by the appropriate delimiter.

- DCB information (see Section 2.5.3), if supplied, must come from the program since the BLKSIZE parameter on the DD statement is ignored.
- If the processing program gives no DCB information, RECFM=F, LRECL=80, and BLKSIZE=80 are assumed.
- A cataloged procedure cannot contain a DD * or a DD DATA statement.
- If a group of data is included and no DD * or DD DATA statement appears, the system provides a DD * statement having SYSIN as its ddname.
- If the delimiter specified in DLM=aa contains any special characters, it should be enclosed in apostrophes.

Examples

```
//GO.SYSIN DD *
           .
           INPUT DATA
           .
/*
//SORTIN DD DATA,DLM=QQ
          .
          INPUT DATA
          .
QQ
```

2.5.2 Output Stream Data Set Parameters (SYSOUT,COPIES,DEST)

The parameter SYSOUT is used to direct a data set to the printer or remote tape unit.

```
SYSOUT=(class,,form),COPIES=c,CHARS=(ttnn,ttnn),DEST=d,DCB=
dcb code
```

For details on the many printing options and how they interact, see the *NIH Computer Center User's Guide*.

Usage at NIH

- "class" which must be given, can have the following values:

A	indicates that output is to be printed.
C	indicates that data is to be transmitted to a remote tape unit.
n	a digit (0-9) indicates that output is to be printed and allows WYLBUR's FETCH command to select data sets as a group if the job is in OUTPUT HOLD.
J	suppresses the printing of selected output.
- The second subparameter in the SYSOUT= field is not supported at NIH. Therefore, if the third subparameter ("form" or "code") is used, a comma must be included to dummy out the second subparameter (i.e., indicate that there is no second parameter).
- Not all printing options are available in the SYSOUT DD statement. Therefore, it is recommended that the /*DDOUT facility, which permits specification of all possible printing options, be used instead.
- The values for forms, copies, characters, and destination are overridden by any given on a /*DDOUT statement. The values for forms, characters, and copies override any given in a /*JOBOUT statement. Values for printing single- or double-sided cut-sheet output on the cut-sheet laser printer are given on an OUTPUT JCL statement.
- "form" is the type of paper to be used for printing the data set at the central facility or remote printer. Forms available at the central facility are discussed fully in the *NIH Computer Center User's Guide*.
- "d" specifies the print routing; this may include central, remote printing, or a VPS node. Rn indicates the number of a remote; CENTRAL may also be given. An invalid remote number may result in lost output. The default is to direct the output to the source location of a job, i.e., the remote that submitted the job, or CENTRAL if the job was submitted via WYLBUR or TSO or through the central facility.

Basic Coding Information

- | | |
|--------|--|
| "form" | can specify any 1 to 4 alphabetic or national character string that is the name of an output form. If a form name is specified that is not recognized at a given printer, the output may not be printed or may be printed on another form. |
| "code" | specifies a 1 to 4 alphanumeric or national character string which identifies |

the /*DDOUT statement that will provide the output characteristics. See Section 2.11.3.

- "c" specifies the number of copies of the SYSOUT data set to be produced; the default is 1. A maximum of 250 copies is allowed, provided the maximum lines specified for the job is not exceeded.
- "ttnn" specifies the character sets (or fonts) to be used to print the data set. Character sets are described in the *NIH Computer Center User's Guide*. Parentheses are not needed when only one font is specified.
- "dcb" specifies the DCB characteristics of the output data set.

- Parentheses are not needed when only the output class is specified.
- SYSOUT data sets are printed in the job output in the order that the DD statements are coded. Therefore, the order of this output can be altered by changing the order of the DD statements.

Additional Information

- When carriage control is specified for the SYSOUT=A data set, column 1 of the data will be used for carriage control and column 2 through the end of the record will be printed on one line. If a record is too long to fit on one line on the form being used, it will be truncated.
- SYSOUT=C output is counted against the job's print limit. The lines-per-job field of the JOB statement should reflect the total output to both SYSOUT=A and SYSOUT=C.
- Multiple destinations cannot be specified on the DD statement. They can, however, be stated on the /*DDOUT statement (see Section 2.11.3).
- The WYLBUR LOCATE and TSO STATUS (or LOCATE) commands give information for each data set waiting output processing.

Examples

```
//OUT          DD  SYSOUT=A

//GO.FT02F001  DD  SYSOUT=(A,,999),
//              COPIES=2,CHARS=(SN12,SE12),
//              DCB=(RECFM=FBA,LRECL=133,
//              BLKSIZE=6251,OPTCD=J)

//GO.COPY      DD  SYSOUT=(A,,FFF)

//PRINTER      DD  SYSOUT=A,DEST=(NIHEPS,WXYZPS)

//OUT1         DD  SYSOUT=J
```

2.5.3 DCB (Data Control Block) Parameter

The DCB parameter provides information about the attributes of a data set.

```
DCB=(list-of-attributes)
      dsname
      *.ddname
      *.step.ddname
      *.step.procstep.ddname
```

Usage at NIH

The following DCB subparameters (attributes) are not supported:

BUFIN	FUNC
BUFMAX	GNCP
BUFOUT	INTVL
BUFSIZE	PCI
CODE	RESERVE
CPRI	STACK
FRID	THRESH

- DSORG values ending in U create unmoveable data sets which are not permitted at this installation (e.g., DSORG=PSU is not allowed).

Basic Coding Information

"list-of-attributes" items DCB subparameters used to describe a data set. Since they are DCB keyword subparameters, they can be specified in any order. The following subparameters are described in Section 2.5.4:

BLKSIZE	LRECL
BUFNO	OPTCD
DSORG	RECFM

Additional DCB subparameter for tapes (see Section 2.5.5):

DEN (9 track)

For additional information on these and other subparameters, see the IBM manual, *OS/390 MVS JCL Reference*, GC28-1757.

"dsname" the name of a cataloged data set from which DCB information is to be copied. The cataloged data set must reside on an online direct access volume.

*.ddname" tells the system to use the DCB associated with the named DD statement which appears earlier in this job step.

"*.step.ddname"	tells the system to use the DCB associated with the named DD statement, which appears in an earlier step named "step" in this job.
"*.step.procstep.ddname"	tells the system to use the DCB associated with a DD statement in a previously executed procedure.
"step"	the name of the job step, which executes the procedure.
"procstep"	the name of the procedure step which includes the DD statement, and "ddname" is the name of the DD statement.

- Parentheses are not needed if only one keyword subparameter, a cataloged data set name, or a backward reference (*.) is used.

Additional Information

- Every data set has a data control block associated with it.
- The data control block is originally constructed from within a program.
- The system may obtain data set attributes (for the data control block) from three places:
 - From the processing program which constructs the data control block.
 - From the DCB parameter on the DD statement.
 - From information recorded in the data set label.

These sources of information are merged when the data set is opened. The program information is used first, then the DD statement, then the data set label.

Examples

```
//GO.OUTPUT1 DD DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440),
//           DISP=(MOD,KEEP),DSN=ABC1XYZ.DATA

//GO.OUTPUT2 DD DCB=*.OUTPUT1,UNIT=FILE,DISP=(NEW,CATLG),
//           DSN=ABC1XYZ.MASTER,SPACE=(CYL,(15,1),RLSE)
```

2.5.4 DCB Subparameters

This section discusses the commonly used DCB subparameters DCB, LRECL, and RECFM, as well as the infrequently used DSORG, BUFNO, and OPTCD.

2.5.4.1 BLKSIZE, LRECL and RECFM

DCB subparameters blocksize (BLKSIZE), logical record length (LRECL), and record format (RECFM) are required fields in the data control block for most applications.

DCB=(BLKSIZE=physical-bytes,LRECL=logical-bytes,
RECFM=record-format)

Usage at NIH

- Small blocksizes (e.g., BLKSIZE=400) can severely degrade processing efficiency for large data sets.
- Blocksizes close to but not exceeding 11476 are recommended for large data sets.
- Recommended blocksizes for commonly used "record" (LRECL) values are:

LRECL	BLKSIZE
80	11440
100	11400
121	11374
133	11438
500	11000

These recommendations apply only to large data sets. WYLBUR's ENTER DISKCALC command provides a convenient way to calculate the optimal blocksize for data sets in the FILE, TMP, and MSS management classes and data sets stored on dedicated disk volumes.

- When a data set is accessed by a batch job, that job must provide sufficient region for up to five buffers for each data set. Since each buffer must be the same size as the data set's BLKSIZE, considerable region may be required. It is important to balance the cost of the large region needed for 2 large blocksizes against the storage efficiency offered by such large blocksizes. Generally, if a data set is fairly small (2-3 tracks), it may be more cost effective to use blocksizes that are significantly smaller than those recommended above.
- Very large blocksizes hamper tape hardware and software error detection and correction mechanisms.
- Blocksize must be compatible with all software that is to access the data set.

Basic Coding Information

"physical-bytes" the maximum number of bytes in a block (physical record). This value may not exceed 32760.

"logical-bytes" specifies the actual length, in bytes, for fixed-length records. For variable length records, "logical-bytes" must be 4 bytes larger than the maximum data record within the data set.

"record-format" specifies the format and characteristics of the data set records. The three basic formats are:

F Records are of fixed length

U	Records are of undefined length
V	Records are of variable length

- If the data set blocksize is not specified by the processing program or on the data set label, the BLKSIZE subparameter must be used.
- During a job's conversion process, testing is done to insure that valid options are specified for the RECFM, but no verification is made for the combination of such parameters. For example, an invalid combination such as RECFM=UV will not be detected during the conversion or interpretation processes.
- All supported access methods except BDAM allow blocking of fixed length and variable length records. In this case "record-format" would be one of the following:

FB	Fixed length records, blocked
VB	Variable length records, blocked

For variable blocked records, the value given by BLKSIZE must be at least 4 bytes larger than the LRECL specified. If the longest record is 200 bytes, then the LRECL must be given as 204 and the blocksize must be 208 or greater.

- The sequential access methods, BSAM and QSAM, allow "S" to be specified as a characteristic of fixed length or variable length records. For variable length records (RECFM=VS or RECFM=VBS), this indicates that a logical record may span more than one block. For fixed length records, (RECFM=FBS), this indicates that the records are written as standard blocks (no truncated blocks, except the last).

If a data set being read FBS has a short block in the middle (as it may if it was extended with DISP=MOD), it cannot be read properly. Attempting to read a data set as FBS specifying a blocksize larger than the one used to create it causes the first block to be treated as a short block.

- The BPAM, BSAM and QSAM access methods allow the RECFM to indicate that the records contain printer control characters. The letter "A" designates ASA control characters. The letter "M" designates machine control characters. The following combinations are valid:

FA	FBA	FBM
FBSA	FBSMFM	
UA	UM	VA
VBA	VBM	VBSA
VBSM	VM	VSA
VSM		

- The "T" designation to indicate track overflow (e.g., RECFM=FBT) is not applicable for NIH Computer Center disks, because the size of a track is greater than the maximum allowable blocksize.

Additional Information

- Excluding the BLKSIZE subparameter in the DCB parameter tells the operating system to calculate the optimum BLKSIZE for a given LRECL (record length).
- An alternative to excluding the BLKSIZE is to specify BLKSIZE=0 which will also cause the operating system to determine the optimum blocksize. This provides a way to use this technique even for batch cataloged procedures in which the blocksize is passed as a symbolic parameter on the EXEC statement.)
- When a data set is created with the system determining the optimum blocksize, it will be automatically reblocked to a new optimum blocksize if the data set has to be moved to a new DASD device with different track/cylinder architecture.

Examples

```
//OUTRECS      DD DCB=(RECFM=FB,LRECL=100,BLKSIZE=11400),
//              UNIT=TAPE,VOL=(PRIVATE,SER=654321),
//              DISP=(NEW,PASS),DSN=ABC1XYZ.JULY.REPORT

//GO.FT09F001  DD DCB=(RECFM=VBS,
//              LRECL=9000,BLKSIZE=6000),
//              UNIT=TMP,DISP=(NEW,CATLG),
//              DSN=ABC1XYZ.DATA,SPACE=(TRK,(10,5),RLSE)

//STEP1 DD      DSN=ABC1XYZ.TESTIT,UNIT=FILE,
//              DISP=(NEW,KEEP),SPACE=(TRK,(1,1,10),RLSE),
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)

//OUTDATA      DD DCB=(RECFM=FB,LRECL=100,BLKSIZE=11400),
//              UNIT=MSS,DISP=(NEW,CATLG),
//              SPACE=(CYL,(10,2),RLSE),
//              DSN=ABC1XYZ.RECS
```

2.5.4.2 DSORG, BUFNO and OPTCD

The DSORG (data set organization), BUFNO (buffer number), and OPTCD (option code) subparameters are covered in this section.

```
DCB=(DSORG=organization,BUFNO=n,OPTCD=code)
```

Usage at NIH

- The organizations PSU, DAU, ISU and POU cannot be used at this installation because such data sets are unmoveable. This makes disk maintenance, which involves condensing and copying data sets, impossible.

- IS organization (ISAM) is not supported by the NIH Computer Center and is therefore not allowed on public disks. Use of ISAM is strongly discouraged, and the NIH Computer Center recommends that any remaining ISAM applications be converted to VSAM.

Basic Coding Information

"organization" indicates the data set organization. If the DSORG subparameter is not provided, a DSORG of PS is assumed.

PS Physical sequential
DA Direct
PO Partitioned data set (PDS)

VS (for VSAM data sets) is not needed and should not be given. The use of VSAM is fully described in the following manuals: *Using Virtual Storage Access Method (VSAM) and Access Method Services at NIH, DFSMS/MVS Access Method Services (ICF)*, SC26-4906, and *DFSMS/MVS Using Data Sets*, SC26-4922.

"n" specifies the number of buffers to be assigned to the DCB. The defaults are 5 for QSAM and 2 for all other access methods.

"code" specifies optional services to be performed by the control program. Many codes are possible; four are of particular interest:

B disregard EOF (end of file) labels on tape; treats multi-volumes as a single data set.

J indicates the output contains Table Reference Characters (TRCs) to print with more than one character set.

Q translation to or from ASCII is required.

Z requests reduced error recovery for magnetic tape input; good for problem tapes.

Refer to the appropriate language manual for information on when DSORG must be coded. For further information, see *OS/390 MVS JCL Reference*, GC28-1757.

Additional Information

- The number of buffers chosen affects the region size required for execution. For a small data set with a large blocksize, fewer buffers will require less region. For large data sets, especially those with small blocksizes, the number of buffers should not be reduced because execution time will be increased.

Examples

```
//LITFILE      DD DCB=(BUFNO=2,OPTCD=Z,
//              RECFM=FB,LRECL=1000,
//              BLKSIZE=11000,DEN=3),
//              VOL=(PRIVATE,SER=123456),
//              UNIT=9TRACKHI,DISP=OLD,DSN=ABC1XYZ.BADTAPE

//DIRECT      DD DCB=(DSORG=DA,RECFM=F,LRECL=500,
//              BLKSIZE=500),
//              UNIT=FILE,SPACE=(TRK,(10,1),RLSE),
//              DISP=(NEW,CATLG),DSN=ABC1XYZ.BDAM

//OUTPUT      DD SYSOUT=(A,,999),CHARS=(SN10,SE10),
//              DCB=OPTCD=J
```

2.5.5 Additional DCB Subparameters for Tapes

The density (DEN) subparameter is discussed in this section. DEN is used only with 9-track tapes; it does not apply to cartridge tapes.

DCB=(DEN=density)

Usage at NIH

- For a complete description of the tape attributes supported by the NIH Computer Center see the *NIH Computer Center User's Guide*.
- DEN should be coded for all 9-track tapes.
- The NIH tape drives insure that any data set added to a tape is written with the same density and parity as the existing data. Any specification of other values will be ignored.

Basic Coding Information

"density" a number which indicates the number of bits-per-inch (BPI) used to write a data set.

DEN=3 indicates 1600 BPI (9-track)

DEN=4 indicates 6250 BPI (9-track)

Examples

```
//GO.INPUT    DD DCB=(RECFM=FB,LRECL=100,
//              BLKSIZE=6200,DEN=3),
//              UNIT=9TRACKHI,VOL=(PRIVATE,SER=012345),
//              LABEL=(1,NL),DISP=(OLD,KEEP),
//              DSN=ABC1XYZ.IN1
```

2.5.6 DDNAME Parameter

The DDNAME parameter allows the description of the data set to be specified in a subsequent DD statement within the same job step.

DDNAME=ddname

Basic Coding Information

"ddname" specifies the name of the DD statement on which the data set will be defined.

- If the definition of a data set is postponed by coding the DDNAME parameter and the ddname is not defined later in the job step, the DD statement is treated as if the DUMMY parameter were specified, and a warning message is issued.

Additional Information

- In any backward reference to a data set, the ddname of the DD statement containing the DDNAME parameter must be used, not the ddname specified in the DDNAME parameter.

Examples

```
//INPUT      DD DDNAME=SYSIN
//OUTPUT     DD SYSOUT=A
//SYSIN      DD *
```

2.5.7 DISP Parameter

The DISP parameter describes: the status of the data set at the beginning of the job step, the disposition of the data set if the step terminates normally, and the conditional disposition of the data set if the step terminates abnormally (ABENDs).

```
DISP=(NEW,DELETE,DELETE)
      OLD  KEEP    KEEP
      SHR  PASS    CATLG
      MOD  CATLG   UNCATLG
           UNCATLG
```

Usage at NIH

- DISP=SHR should be specified for all JOBLIB and STEPLIB DD statements. If a value other than SHR is specified, it will be treated as SHR. Omitting the DISP parameter will cause the job to fail.
- All other DD statements for dsnames prefixed by NIH. or SYS1. should also specify SHR. If the DISP is omitted or another value is specified, it will still be treated as SHR.

Basic Coding Information

- | | |
|-----------|---|
| "NEW" | indicates that the data set is created in this job step. New data sets in the FILE, TMP, and MSS management classes will be cataloged automatically. |
| "OLD" | indicates that the data set existed before this job step and that the step requires exclusive (non-shared) use of the data set. |
| "SHR" | indicates that the data set existed before this job step and can be used simultaneously (shared) by another job, since it will only be read. |
| "MOD" | indicates that the read-write mechanism will be positioned after the last record of the data set and the step requires exclusive (non-shared) use of the data set. If no volume information is available, the data set will be created. |
| "DELETE" | indicates that the data set is to be deleted at the end of the job step. This will also uncatalog the data set if it is referenced through the catalog. |
| "KEEP" | indicates that the data set is to be kept at the end of the job step. |
| "PASS" | indicates that the data set is to be passed to a later job step. |
| "CATLG" | indicates that the data set is to be cataloged and kept at the end of the job step. |
| "UNCATLG" | indicates that the data set is to be uncataloged and kept at the end of the job step. |
- The default for the first parameter is NEW. The default for the second parameter is DELETE for a new data set and KEEP for a data set that existed before execution of the job. The third parameter (for abnormal termination) defaults to the same as the parameter for normal termination disposition (the second parameter).
 - All subparameters are positional; therefore, omitted subparameters that are followed by other subparameters must be replaced by a comma. Parentheses are not needed when only the first subparameter is specified.

Additional Information

- The DISP parameter is ignored if:
 - The job abnormally terminates before the step begins execution.
 - The job step is bypassed.
 - A request for a tape volume is made, and the data set is never opened.

- The data set is defined as a dummy data set.
- Coding a disposition of PASS for a data set causes the UNIT, VOLUME, and DSNAME information to be stored in a pass table. Specifying the data set name of a PASSEd data set without specifying VOLUME or UNIT information is called "receiving" the data set. A data set that is created and PASSEd, but never received, will be deleted at the end of the job.
- A first subparameter of NEW, OLD, or MOD requests exclusive control of the data set and the data set name for the duration of the job step.
- DISP=SHR should be used whenever a data set is only read. This will reduce unnecessary system delays when more than one job attempts simultaneous exclusive access to a data set.
- Temporary data sets must specify a disposition of PASS or DELETE. Anything else will be overridden by the system with PASS.
- Disposition processing for unopened data sets varies, depending on the disposition coded. For specific information, see the manual *OS/390 MVS JCL Reference*, GC28-1757.
- In a Class C job, if (NEW,CATLG) is specified, the tape will be assigned only if the data set was successfully cataloged.
- In order to ensure that all public disk data sets created in batch are automatically cataloged, any JCL specifying (NEW,KEEP) is changed internally by the system to (NEW,CATLG). JCL that specifies (MOD,KEEP) will be automatically changed to (MOD,CATLG). As a result, new data sets that are created via the disposition (MOD,KEEP) will be cataloged. When existing data sets are referenced with the disposition (MOD,KEEP), which is changed automatically to (MOD,CATLG), the message NOT RECTLGD 2 will be displayed in the JES2 job log. This message is informational and does not affect the functioning of the job.

Examples

```
//GO.INPUT DD DISP=SHR,DSN=ABC1XYZ.NAMES,UNIT=FILE,
//          DISP=(NEW,CATLG,DELETE)

//SORTOUT DD DISP=(NEW,PASS),UNIT=SYSDA,DSN=&&XYZSRT,
//          SPACE=(TRK,(100,10),RLSE)

//SORT DD DISP=(OLD,DELETE),DSN=&&XYZSRT

//OUT DD DISP=(NEW,CATLG),DSN=ABC1XYZ.DATA,
//      UNIT=MSS,SPACE=(CYL,(5,1),RLSE)

//NEWMAS DD DISP=(NEW,KEEP,DELETE),
//          DSN=ABC1XYZ.MASTER,UNIT=TAPE,
//          VOL=(PRIVATE,SER=012345)
```

2.5.8 DSNNAME Parameter

The DSNNAME parameter specifies the name of a data set. If the data set is new, the name is assigned to it. If the data set already exists, the name is used to locate it.

DSNNAME=dsname or DSN=dsname

Usage at NIH

- The dsname of any data set that is stored on public volumes or the MSS must follow standard IBM data set naming conventions; in addition, the first eight characters must be the user's registered account and initials followed by a period. For example,

aaaaiii.name

- where "aaaaiii" is the user's registered account/initials. Data sets having dsnames not of this form will be deleted from public volumes. The names of data sets on other devices should also follow this convention.
- For VSAM data sets, an alternate index for a VSAM cluster must reside under the same account/initials combination as the VSAM cluster. That is, both the VSAM cluster name and the name of the alternate index must start with the same account/initials. Data and index component names are generated by the system based upon the cluster name.
- The following special function prefix that precedes the account and initials should be used only for the indicated purpose:

MICRO105.aaaaiii indicates a data set on tape to be processed onto
105 mm microfiche.

- The prefixes SYS1. and NIH. are reserved for system data sets. No attempts should be made to create, modify, catalog, or delete any data sets which have these prefixes.
- Duplicate data set names are not permitted on public disk storage.

Basic Coding Information

- Permanent data sets should follow IBM data set naming conventions with two or more groups of one-to-eight characters, separated by periods. The first character of each group must be alphabetic or national (#, @, \$); the remaining characters can be alphabetic, numeric, national or a hyphen. The data set name, including the periods, must not exceed 44 characters in length.
- If a data set name does not follow all of the above conventions, it must be enclosed in apostrophes (single quotes). For a utility which defines the data set name via a symbolic parameter (i.e., NAME=dsname), the data set name must be enclosed within three single quotes (e.g., NAME="INVALIDNAME"). For straight JCL the data set name must be enclosed in single quotes (e.g., DSN='INVALIDNAME'). If you do not use the appropriate method, your job will terminate with a JCL error. Because more and more software from IBM and other vendors requires use of "standard" data set names, users should make it a practice to use standard dsnames.

- Data set names specified for any new data set on a public volume may not use single quotes, or a JCL error will result. Single quotes are permitted when referencing data sets on tapes, and are required for data set names that do not adhere to standard naming conventions.
- The dsname can have several different forms:

dsname	identifies a data set name.
dsname(member name)	identifies a member of a partitioned data set.
dsname(generation)	identifies a generation data set by its number within a generation data group.
dsname	identifies a temporary data set.
&&dsname(member name)	identifies a member of a temporary partitioned data set.
*.ddname	indicates that the data set name is to be copied from the named DD statement, which must precede this statement in this job step.
*.step.ddname	indicates that the data set name is the same as the data set in the named DD statement which appears in an earlier step named "step" in the same job.
*.step.procstep.ddname	indicates that the data set name is the same as the data set name in an earlier DD statement in a cataloged procedure. "step" is the name of the job step that calls the procedure. "procstep" is the name of the procedure step that includes the named DD statement, and "ddname" is the name of the DD statement that contains the data set name.

Additional Information

- A temporary data set name is preceded by 1 or 2 ampersands (&) and can consist of 1 to 8 alphanumeric characters. The first character after the ampersand must be an alphabetic or national character.
- If a data set with a data set name longer than 17 characters is written on a standard labeled tape, only the 17 right-most characters of the data set name will be recorded on the label.

- Only temporary data sets (a name beginning with & or &&) may be allocated to scratch disks. Jobs that attempt to create permanent data sets on scratch disks will be cancelled.
- The dsname need not be coded if the data set is temporary.
- DSN=NULLFILE specifies a dummy data set. This produces the same effect as DD DUMMY (described in Section 2.5.9). If a user's job opens a dummy data set, the same costs, except for I/O, are incurred by the job as for opening a non-dummy data set.

Examples

```
//GO.FT08F001    DD DSN=ABC1XYZ.INPUT,DISP=SHR

//LOAD.SYSLIN    DD DSN=ABC1XYZ.LOAD(MAIN),DISP=SHR

//TEMPIN        DD DSN=&&XYZTMP,UNIT=TAPE,
//              DISP=(NEW,PASS)

//GO.SORTOUT     DD UNIT=SYSDA,DISP=(NEW,PASS),
//              SPACE=(TRK,(100,10),RLSE)

//SORTED        DD DSN=*.GO.SORTOUT,DISP=(OLD,DELETE)

//STEP1         EXEC DSGET,NAME=' 'INVALIDNAME' ',
//              TAPE=123456,DRIVE=9TRACKHI

//COPY.SYSUT1    DD DSN='INVALIDNAME',
//              UNIT=9TRACKHI,DISP=(OLD,KEEP),
//              VOL=(PRIVATE,SER=ABC123)
```

2.5.9 Dummy Parameter

The DUMMY parameter bypasses the processing of a data set.

```
DUMMY
```

Basic Coding Information

- The DDNAME parameter may not be used with the DUMMY parameter.
- The DCB parameter may be used to supply data set control block information for a "dummy" data set.
- Any parameters (other than DDNAME and DCB) which are coded with the DUMMY parameter will be checked for syntax but otherwise ignored.
- When a program reads a data set defined with the DUMMY parameter, an end-of-data-set exit is taken immediately.
- When a program requests that a "dummy" data set be written, the request is recognized, but no data is transferred.

- If the "dummy" data set is opened, the user's job incurs the same costs, except for I/O, that opening a non-dummy data set would incur.

Examples

```
//OUTPUT      DD DUMMY,DCB=(RECFM=FB,LRECL=80,
//              BLKSIZE=11440)
```

2.5.10 LABEL Parameter

The LABEL parameter indicates the label type and the relative position of the data set on the tape and whether the data set is to be opened for input or output.

```
LABEL=(seqno,SL,,IN)
        AL  OUT
        NL
```

Usage at NIH

- Only SL, NL and AL labels are supported.
- The following subparameters are not supported:

```
PASSWORD      RETPD
NOPWREAD      EXPDT
```

Basic Coding Information

"seqno" the data set sequence number, specifies the relative position of the data set on a tape volume.

"SL" indicates that the tape has IBM standard labels.

"AL" indicates that the tape has American National Standard (ANSI) labels.

"NL" indicates that the tape has no labels.

"IN" specifies that a data set is to be used for input only.

"OUT" specifies that a data set is to be used for output only.

- The LABEL parameter is ignored for disk data sets.
- The LABEL parameter must be used if any of the following applies:
 - The tape data set is not the first data set on the volume ("seqno" must be supplied).
 - The tape data set does not have IBM standard labels (NL or AL must be coded).
- All subparameters are positional; therefore, omitted subparameters that are followed by other subparameters must be replaced by a comma. Parentheses are not needed if only the first subparameter is specified.

- BLP (Bypass Label Processing) is not supported. Any type that needs to be handled by bypassing the labels (e.g., a damaged tape) should be processed using PALTAPE (see 3.5.3).

Additional Information

- The data set sequence number has a default of 1.
- The label type has a default of SL.
- Label type information is not retained for cataloged or passed data sets; therefore, it must be specified if anything other than SL is required.
- LABEL=(n,AL) can be specified for an output ASCII data set only if the ANSI labeled output tape was originally created at an installation that supports "Version 3" ISO/ANSI standards. Refer to the *NIH Computer Center User's Guide*.
- The IN and OUT subparameters are used only with the BSAM and BDAM access methods.

Examples

```
//OUTPUT          DD LABEL=2,DSN=ABC1XYZ.JOB.STATS.TWO,
//                UNIT=TAPE,VOL=(PRIVATE,SER=123456),
//                DISP=(MOD,KEEP)

//GO.FT10F001     DD LABEL=(3,,IN),DSN=ABC1XYZ.DATA,
//                UNIT=TAPE,VOL=(PRIVATE,SER=789123),
//                DISP=SHR

//STATS           DD LABEL=(1,NL),DSN=ABCD1XYZ.STATS,
//                UNIT=TAPE,VOL=(PRIVATE,SER=456789),
//                DISP=(OLD,KEEP),
//                DCB=(RECFM=FB,LRECL=80,BLKSIZE=8000)
```

2.5.11 SPACE Parameter

The SPACE parameter is used to request space on a disk volumes.

```
SPACE=(TRK,(primary,secondary,directory),RLSE,CONTIG)
      CYL
      blocklength
      reclgth
```

Usage at NIH

- The ABSTR subparameter of SPACE is not supported at NIH.

Basic Coding Information

"TRK" specifies that space is to be allocated in units of tracks.
 "CYL" specifies that space is to be allocated in units of cylinders.

"blocklength"	specifies that space is to be allocated in units of average block length. The system computes the number of tracks to allocate. This cannot be used if AVGREC is coded.
"reclgth"	specifies the average record length of the data in bytes. The system computes the blocksize and how many tracks to allocate. Use this only if AVGREC is coded.
"primary"	specifies the number of tracks, cylinders, or blocks required to store the data set. When the AVGREC parameter is coded, primary represents the number of records. For a PDS, the number of records is the total number of records in all members.
"secondary"	specifies the number of tracks or cylinders to be allocated if additional space is required. This additional space should reflect the expected growth of the data set. (Five to ten percent of the "primary" value is a safe value.)
"directory"	specifies the number of 256-byte directory blocks to be contained in the directory of a partitioned data set.
"RLSE"	indicates that space not used by the data set is to be released; RLSE should be specified whenever possible.
"CONTIG"	specifies that the primary space allocated must be contiguous.

- If "secondary" and "directory" are omitted, the parentheses around primary are not needed. A comma must be coded if the secondary is omitted and the directory is coded. Similarly, a comma must be coded if RLSE is omitted and CONTIG is coded.

Additional Information

- The SPACE parameter is ignored for tapes.
- The SPACE parameter must be provided when any data set is created on a disk. The *NIH Computer Center User's Guide* describes how to estimate the space requirements for a data set using ENTER DISKCALC.
- The NIH Computer Center suggests that you allocate, as close as possible, the amount of space required in "primary" and use the "RLSE" (release) option. "Secondary" space should be 10 %of the "primary" allocation. If no "secondary" is specified, unused space is not released.
- Space on the public volumes can be allocated in tracks, cylinders, blocks, or average record length of the data in bytes.
- There must be enough available space on one volume, in up to 5 noncontiguous areas (extents), to satisfy the primary quantity; otherwise, the job will be terminated.
- If a secondary quantity is required, there must be enough space in up to 5 noncontiguous areas (extents) to satisfy each request for the secondary quantity. Additionally, no more than 16 noncontiguous areas (extents) will be used to satisfy the entire SPACE request (primary and total secondary). If this space is not available, the job will ABEND.

- Expansions of existing data sets use the secondary quantity to allocate more space. If expansion is anticipated, the SPACE request should allow for expected growth, and RLSE should not be coded. BDAM data sets cannot be expanded once they are created.
- If an expansion request would cause the data set to be larger than allowed on the volume class, the request will be reduced, and the expansion will succeed. If the data set is already at the maximum size and expansion is requested, the job will fail.
- Directory space must be allocated when a partitioned data set is created. A DSORG parameter of PO must be specified. See Section 2.5.4.2 for additional information.
- For relative block address BDAM data sets, it is important to request only the number of tracks needed because writing of dummy records to fill unneeded space can increase I/O, CPU, and storage costs.
- If CONTIG is specified and the requested contiguous space is not available, the job will be terminated. Contiguous space should not be requested unless it is required by the program. Requesting it reduces the chance of there being suitable space available. Contiguous space will always be allocated if it is available.
- If contiguous space is requested and a secondary quantity is coded, the primary request will be satisfied with contiguous space, but the secondary quantity may not be contiguous.
- Key Length (KL) must be considered for some BDAM data sets when calculating space requirements on a 3390. See the *NIH Computer Center User's Guide* for additional information.
- On public disk volumes, a single data set may not be split between volumes (i.e., multi-volume data sets).

Examples

```
//STEP1      DD SPACE=(TRK,(10,1),RLSE),UNIT=SYSDA,
//           DISP=(NEW,PASS)

//FIRST      DD SPACE=(11440,(100,10),RLSE),UNIT=FILE,
//           DSNAME=ABC1XYZ.FORX.STATS,
//           DISP=(NEW,CATLG),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440)

//ONE        DD SPACE=(CYL,(20,,1),RLSE),
//           UNIT=SYSDA,DISP=(NEW,PASS),
//           DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440,DSORG=IS)
```

2.5.12 UNIT DD Parameter

The UNIT parameter describes the input or output device used by a data set. The UNIT parameter is not needed when referencing a cataloged data set and no VOLUME parameter is specified.

UNIT=(device,count,DEFER) or UNIT=AFT=ddname
 P

Usage at NIH

- Device names are listed below:

UNIT=TAPE	specifies a drive capable of processing the NIH Computer Center's standard tape.
UNIT=CTAPE	specifies a drive capable of processing a "special" 3480 cartridge tape.
UNIT=9TRACKHI	specifies a drive capable of processing 9-track 6250 or 1600 BPI tapes.
UNIT=FICHE	specifies a drive capable of processing tapes to be used for microfiche output.
UNIT=SYSDA	specifies a system scratch disk (for storage of temporary data sets only).
UNIT=FILE	specifies the FILE management class on a public volume or dedicated (private) disk.
UNIT=TMP	specifies the TMP management class on a public volume.
UNIT=MSS	specifies the MSS management class on a public volume.

Basic Coding Information

"device"	specifies a group of drives where the volume is (or will be) mounted.
"count"	indicates the number of drives to be assigned to the data set. The use of count is never recommended except on system scratch disk volumes (see "Additional Information").
"P"	indicates a parallel mount is desired (i.e., each data set is assigned a different device so that all the tapes are mounted at the same time). The use of P is never recommended (see "Additional Information").
"DEFER"	specifies that a drive should be assigned, but the volume will not be mounted until the data set is opened.
"AFF"	indicates that within a job step, different data sets residing on different tapes can be allocated to the same unit.
"ddname"	specifies the name of an earlier DD statement in the job step that defines the data set with which unit affinity is desired.

- All subparameters are positional; therefore, omitted subparameters that are followed by other subparameters must be replaced by a comma. Parentheses are not needed when only the first subparameter is specified.

Additional Information

The UNIT parameter must always be coded except when:

- The VOLUME parameter is not specified, and the data set was passed from a previous step.
- The data set is cataloged, and the VOLUME parameter is not specified.
- VOL=REF=reference is coded.

-
- The data set has SYSOUT=A or C specified.
 - DUMMY, *, or DATA is used to define the data set.
 - If a data set is passed from a previous step, the UNIT and VOLUME parameters should not be specified.
 - If the data set is cataloged and the VOLUME parameter is not coded, the system will obtain unit and volume information from the catalog. If UNIT is specified, it must be compatible with the catalog information and valid for the data set. However, if the volume serial number is coded, UNIT must be supplied because the catalog will not be searched.
 - The following shows the effect of specifying different unit types when allocating new data sets nonspecifically (i.e., without specifying a VOL=SER=parameter):

UNIT=FILE	The system chooses the FILE management class on public storage.
UNIT=TMP	The system chooses the TMP management class on public storage.
UNIT=MSS	The system chooses the MSS management class on public storage.
 - If VOL=REF=reference is coded (see Section 2.5.13), the data set uses the same volume assigned to an earlier data set. The system obtains unit and volume information from the earlier DD statement.
 - If DEFER is specified and the data set on that volume is never opened, the volume is not mounted even though the UNIT is allocated. However, this will save an unnecessary tape mount if the tape may not be needed depending on what happens in other steps. No other jobs can use that volume until the step specifying DEFER ends, and any attempt to use such a volume by other jobs will cause all but the first job to enter a "wait" state.
 - It is strongly recommended that multi-volume tape data sets use only one drive, rather than specifying "count" or "P" to use multiple drives. The use of these parameters will not generally affect the execution time of an individual job; however, it may degrade performance of other jobs in the system.
 - To obtain the maximum allowable space for a temporary sequential or direct data set on system scratch volumes, it may be necessary to code UNIT=(SYSDA,2).
 - Two data sets on different volumes can be assigned to the same device by specifying UNIT=AFF=ddname, where "ddname" defines the DD statement of the data set to share the drive.
 - For tapes, specifying a unit, such as 9TRACKHI, does not imply an equivalent density. The DENSITY parameter, described in Section 2.5.5 must also be given.
 - Any job requesting a tape must include a /*ROUTE XEQ resource statement to insure it runs on a subsystem with the needed resource.

Examples

```
//SORTWK01      DD UNIT=SYSDA,SPACE=(CYL,(80),RLSE)

//GO.FT10F001   DD UNIT=(TAPE,,DEFER),DISP=(OLD,KEEP),
//              VOL=(PRIVATE,SER=123456),
//              DSN=ABC1XYZ.INPUT,LABEL=(,,IN)

//INPUT         DD UNIT=TMP,DSN=ABC1XYZ.INPUT,
//              DISP=(NEW,CATLG)

//S5            DD UNIT=MSS,SPACE=(CYL,(5,1),RLSE),
//              DISP=(NEW,CATLG),DSN=ABC1XYZ.DATA

//TAPE          DD UNIT=CTAPE,VOL=SER=SPECL1,
//              DISP=OLD,DSN=ABC1XYZ.DATA
```

2.5.13 VOLUME Parameter

The VOLUME parameter provides information about the tape or disk volume(s) on which a data set resides. The VOLUME parameter is not needed when referencing a cataloged data set.

```
VOLUME=(PRIVATE,RETAIN,seqno,count,SER=(serial,serial...))
      or                                REF=dsname
VOL                                     REF=*.ddname
                                      REF=*.step.ddname
                                      REF=*.step.procstep.ddname
```

Usage at NIH

- All NIH Computer Center tapes have 6 numeric digits as volume serial numbers; zeroes should be inserted to the left to pad to 6 digits if necessary.

"PRIVATE"

indicates that output data sets cannot be allocated to this volume unless the volume is specifically requested. For tapes, the volume will be demounted after its last use in the job step, unless RETAIN is coded or a data set on the volume is passed. PRIVATE should not be used for public online volumes (e.g., the MSS).

"RETAIN"

indicates that the volume is not to be demounted after its last use in the job step. The volume will be kept mounted until its next reference, or the end of the job, whichever comes first. This can reduce the number of tape mounts and the tape mount charges for a job.

"seqno"	indicates which volume of an existing multi-volume data set is to be processed first.
"count"	indicates the maximum number of tape volumes an output data set may require. This must be specified if the data set is to exceed 5 volumes and the specific volume serial numbers are not coded.
"SER=(serial,serial,serial...)"	gives the serial number(s) of the volume(s) on which the data set resides; they must be in order of creation.
"REF=dsname"	tells the system to use the same volumes that were assigned to the named passed or cataloged data set.
"REF=*.ddname"	tells the system to use the same volume(s) that were assigned to the named DD statement, which must precede this statement in this job step.
"REF=*.step.ddname"	tells the system to use the same volume(s) that were assigned to the named DD statement which appears in an earlier step named "step" in the same job.
"REF=*.step.procstep.ddname"	tells the system to use the same volume(s) assigned to the named DD statement in a cataloged procedure. "step" is the name of the job step that executes the procedure. "procstep" is the name of the procedure step that includes the named DD statement, and "ddname" is the name of the DD statement that contains the volume serial number(s).

- The VOLUME parameter must be given to:
 - create a new data set on more than five scratch volumes
 - request an old data set which is not cataloged or passed
 - mount a "scratch" volume which is to be assigned to the user at the end of the job
 - create a new data set on a specific volume
- If the only subparameter coded is PRIVATE, SER=, or REF=, the parentheses are not required.
- If only one serial number is coded, it need not be enclosed in parentheses. All subparameters other than SER are positional; therefore, omitted subparameters that are

followed by other subparameters must be replaced by a comma. SER is a keyword subparameter and must follow all positional parameters.

Additional Information

- If a volume is to be PRIVATE, every DD statement referencing that volume should have PRIVATE coded.
- Existing non-cataloged data sets must have a specific volume request. A specific volume request is implied when:
 - VOL=SER=serial is coded on the DD statement.
 - The data set is passed from an earlier job step or is cataloged.
 - VOL=REF=reference is coded on the DD statement, referring to an earlier specific volume request.
 - New tape data sets may have a specific volume request or allow the system to assign a scratch volume.
- If scratch volumes assigned by the system to a job are to be kept after the job, the job must be run Class C (see Section 2.2.4).
- If UNIT=FILE is specified, the system will search the public volumes to locate one which will satisfy the primary space request and does not contain a data set with the same data set name. The data set created will be in the FILE management class.
- If UNIT=TMP is specified, the system will search the public volumes to locate one which will satisfy the primary space request and does not contain a data set with the same data set name. The data set created will be in the TMP management class.
- When using a PASSED or cataloged data set, do not code any serial numbers or use VOL=REF=reference. Doing so would mean the data set would not be referenced through the system's internal pass table or the catalog.
- If a multi-volume data set is to be MODed in more than one step, the user must be aware of which tapes are written on in each step and specify the sequence number for the tape on which the data set ends.
- VOL=REF=dsname can be used to allocate new data sets on a dedicated pack without giving the specific volume serial number in the JCL. This method eliminates the need to change the JCL in the event that the volser of the dedicated pack changes. VOL=REF should not be used for multi-volume data sets because only the last volume serial number is used.

Examples

```
//GO.PRINT1      DD VOL=PRIVATE,UNIT=TAPE,
//              DSN=ABC1XYZ.PRINT,DISP=(NEW,KEEP)

//GO.FT10F001    DD VOL=(PRIVATE,RETAIN,
//              SER=(123456,012345)),
```

```

//          UNIT=TAPE,DSN=ABC1XYZ.DATA,
//          LABEL=( , , IN),DISP=(OLD,KEEP)

//GO.FT11F001 DD VOL=(PRIVATE,REF=*.GO.PRINT1),
//          DSN=ABC1XYZ.DATA2,DISP=(OLD,KEEP),
//          LABEL=(2 , , IN)

//GO.EZFOUTPT DD DSN=MICRO105.XYZ1ABC.FISH,
//          DISP=(MOD,KEEP,DELETE),
//          VOL=(PRIVATE, , 2),
//          DCB=(RECFM=FBA,LRECL=135,
//          BLKSIZE=3915),UNIT=MICRO

```

In the example below, the data set AAAAIII.NEWDATA will be stored on the dedicated pack on which the data set AAAAIII.DISK1 currently resides.

```

//MYJOB      DD DSN=AAAAIII.NEWDATA,UNIT=FILE,
//          DISP=(NEW,KEEP),VOL=REF=AAAAIII.DISK1,
//          SPACE=(50,(20,2,100),RLSE),AVGREC=M

```

2.5.14 AVGREC Parameter

The AVGREC parameter specifies that the units of allocation requested for storage space for a new data set are records and that the primary and secondary space quantity specified on the SPACE parameter represents units, thousands, or millions of records.

```

AVGREC=U
      K
      M

```

Basic Coding Information

- "U" indicates that the multiplier factor is in units. The primary and secondary values in the SPACE parameter will be multiplied by 1.
- "K" indicates that the multiplier factor is in thousands. The primary and secondary space values will be multiplied by one kilobyte (1,024).
- "M" indicates that the multiplier factor is in millions. The values will be multiplied by one megabyte (1,048,576).

Additional Information

- When AVGREC is used with the SPACE parameter, the first subparameter (reclgth) on the SPACE parameter must specify the average record length of the records. Refer to Section 2.5.11 for a description of the SPACE parameter.
- Excluding the BLKSIZE subparameter in the DCB parameter tells the operating system to calculate the optimum BLKSIZE for a given LRECL (record length).
- An alternative to excluding the BLKSIZE is to specify BLKSIZE=0 which will cause the operating system to determine the optimum block size. This provides a way to use this

technique even for batch cataloged procedures in which the block size is passed as a symbolic parameter on the EXEC statement.)

- When a data set is created with the system determining the optimum blocksize, it will be automatically reblocked to a new optimum block size if the data set has to be moved to a new DASD device with a different track/cylinder architecture.

Examples

Each of the following statements results in the allocation of a partitioned data set (PDS) large enough to hold 20 million 50-character records with a secondary of two million records and 100 directory blocks:

```
// SPACE=( 50 , ( 20 , 2 , 100 ) , RLSE ) , AVGREC=M
// SPACE=( 50 , ( 20000 , 2000 , 100 ) , RLSE ) , AVGREC=K
// SPACE=( 50 , ( 20000000 , 2000000 , 100 ) , RLSE ) , AVGREC=U
```

Note that if this were a variable length (RECFM=VB) data set, 50 (first space parameter) would represent the average length of the records in the data set, not the "LRECL" length.

The following example creates a sequential data set that holds 10,000 records. All records are 80-characters long (e.g., source code for a large COBOL program). The SPACE and AVGREC parameters tell the operating system to calculate the amount of space needed for 10,000 records that are 80-characters long (LRECL) with a secondary space allocation that allows room for 1,000 records. (The RLSE subparameter is recommended in case the estimate of the number of records is too high.)

```
//GO.INPUT DD DSN=ABC1XYZ.MY.DATA.SET,UNIT=FILE,
//          DISP=(NEW,CATLG),
//          DCB=(DSORG=PS,RECFM=FB,LRECL=80),
//          SPACE=(80,(10000,1000),RLSE),AVGREC=U
```

2.5.15 LIKE Parameter

The LIKE parameter is used to specify the allocation attributes of a new data set by copying the attributes of another data set. The model data set must be an existing cataloged data set.

LIKE=data set name

Basic Coding Information

"dsname" is the name of the data set whose attributes are to be used as attributes of the new data set.

Additional Information

- The LIKE parameter is used to copy all the DCB characteristics from the old data set.

- If reblocking, and the new data set requires less space, the SPACE parameter should be respecified.

Examples

The following examples reblock an existing sequential (PS) data set and a partitioned (PO) data set to optimum block sizes.

This example reblocks a sequential data set to optimum blocksize. It requests primary space for 10,000 80-byte records and secondary space for 1,000 80-byte records.

```
//COPY.SYSUT1 DD DSN=ABC1XYZ.POORLY.BLOCKED.DS,DISP=SHR
//COPY.SYSUT2 DD DSN=ABC1XYZ.OPTIMUM.BLOCKED.DS,
//           DISP=(NEW,CATLG),UNIT=FILE,
//           LIKE=ABC1XYZ.POORLY.BLOCKED.DS,
//           SPACE=(80,(10000,1000),RLSE),AVGREC=U
```

This example reblocks a partitioned data set to optimum block size. The example requests primary space for 10,000 80-byte records, secondary space for 1,000 80-byte records, and 10 directory blocks, which provide room for 50-60 members.

```
//COPY.SYSUT1 DD DSN=ABC1XYZ.POORLY.BLOCKED.DS,DISP=SHR
//COPY.SYSUT2 DD DSN=ABC1XYZ.OPTIMUM.BLOCKED.DS,
//           DISP=(NEW,CATLG),UNIT=FILE,
//           LIKE=ABC1XYZ.POORLY.BLOCKED.DS,
//           SPACE=(80,(10000,1000,10),RLSE),AVGREC=U
```

2.5.16 DATACLAS Parameter

The DATACLAS parameter is used to specify a data class for a new data set. The data class are names predefined by the installation and have assigned DCB and SPACE parameter values.

```
DATACLAS=data-class-name
```

Usage at NIH

DATACLAS descriptions for non-VSAM data sets

"DATAF"	defines a fixed block, physical sequential, 80-character data set with default primary and secondary space values of 500 records; enough space to hold up to 8,000 records.
"DATAV"	defines a variable blocked, physical sequential, 255-character data set with default primary and secondary space values of 150 records; enough space to hold at least 2,400 records (more depending on how much the average length of the records is less than the maximum length of 251).

"LISTF"	defines a fixed block, physical sequential, 133-character data set with ANSI carriage control and default primary and secondary space values of 350 records; enough space to hold up to 5,600 records.
"LISTV"	defines a variable blocked, physical sequential, 137-character data set with ANSI carriage control and default primary and secondary space values of 340 records; enough space to hold at least 5,440 records (more depending on how much the average length of the records is less than the maximum length of 133).
"LOADLIB"	defines a partitioned load library into which programs can be resolved using the Binder with default primary space value of 16K bytes and secondary space value of 64K bytes—enough space to hold 61 16K-programs or 30 32K-programs, etc., and 20 directory blocks (sufficient directory space for about 120 members).
"PDSF"	defines a fixed block, 80-character partitioned data set (PDS) with a default primary space value of 250 records and secondary space value of 1,000 records. This is enough space to hold at least 15,250 records total for all members, with 20 directory blocks (sufficient for about 120 members).
"PDSV"	defines a variable blocked, 255-character partitioned data set (PDS) with a default primary space value of 90 records and secondary space value of 350 records—enough space to hold at least 5,340 records total for all members (more depending on how much the average length of the records is less than the maximum length of 251)—and 20 directory blocks (sufficient directory space for about 120 members).

Basic Coding Information

"data class name" a 1-to-8 character name predefined by the installation.

Additional Information

Data classes can be used to simplify the coding of JCL by providing a default for the DCB the SPACE parameters.

- Data classes can be used when allocating data sets on dedicated disks or public disks.
- Specify the data classes in batch jobs by using the DATACLAS JCL parameter, and in TSO via the DATACLAS parameter of the ALLOCATE command. They are only applicable when a data set is being created and will be ignored if used in reference to an existing data set. Data classes only support physical sequential (PS), partitioned (PO), and VSAM (VS) data sets.

-
- The non-VSAM data classes represent the most popular DSORG/ RECFM combinations on the system. The VSAM data classes can be used to allocate VSAM data sets, with some limitations, directly via a batch JCL DD statement instead of executing an IDCAMS DEFINE command (using the VDSUTIL procedure).
 - All space values are specified as records or bytes (instead of tracks or cylinders) making the data classes device independent, i.e., they will be valid even if the current disk storage devices (DASD) on the system are replaced with drives of a different architecture or geometry.
 - The block size defaults to the optimum block size for the device on which the data set is being allocated. The one exception is the LOADLIB data class which has a record format (RECFM) of U (undefined). The block size for data sets created using this data class must be determined by the program writing to them (i.e., the Binder).
 - For VSAM data sets, only the cluster name of the data set is specified in the JCL. The component names are assigned by the system (usually by adding .DATA or INDEX to the end of the cluster name).
 - For VSAM data classes, only the default RECORDSIZE is arbitrary and should be specified via the JCL LRECL parameter (see examples below). Likewise, the KEYS length and offset values for KSDS data sets must be specified using the KEYLEN and KEYOFF JCL parameters.
 - In the VSAM data classes, the RECOVERY, NOERASE, NOREUSE and NONSPANNED default parameters cannot be overridden or modified. If there is a need to create a VSAM data set with the SPEED, ERASE, REUSE, or SPANNED attributes, then the IDCAMS DEFINE command (via the VDSUTIL cataloged procedure) must be used to create the data set. This is also true of the IMBED and REPLICATE attributes of KSDS data sets although the default values of NOIMBED and NOREPLICATE are recommended for devices with cache controllers such as those running at NIH.
 - The following list shows the non-VSAM data classes and their equivalent DCB and SPACE parameters:

DATACLAS PARAMETER	EQUIVALENT DCB AND SPACE PARAMETERS
DATACLAS=DATAF	DCB=(RECFM=FB,DSORG=PS,LRECL=80), SPACE=(80,(500,500)),AVGREC=U
DATACLAS=DATAV	DCB=(RECFM=VB,DSORG=PS,LRECL=255), SPACE=(255,(150,150)),AVGREC=U
DATACLAS=LISTF	DCB=(RECFM=FBA,DSORG=PS,LRECL=133), SPACE=(133,(350,350)),AVGREC=U
DATACLAS=LISTV	DCB=(RECFM=VBA,DSORG=PS,LRECL=137), SPACE=(137,(340,340)),AVGREC=U
DATACLAS=LOADLIB	DCB=(RECFM=U,DSORG=PO), SPACE=(16,(1,4,20)),AVGREC=K
DATACLAS=PDSF	DCB=(RECFM=FB,DSORG=PO,LRECL=80), SPACE=(80,(250,1000,20)),AVGREC=U
DATACLAS=PDSV	DCB=(RECFM=VB,DSORG=PO,LRECL=255), SPACE=(255,(90,350,20)),AVGREC=U
Note:	The above space allocations are equivalent (TRK,(1,1)) for the 4 PS data classes and (TRK,(1,2,20)) for the 3 PO data classes when using our current disk storage devices (3390s).

- The following figure shows the default attributes of the four VSAM data classes:

Figure 2-4. Default Attributes for VSAM Data Classes

ATTRIBUTE	VSAM DATA CLASSES			
	VSAMESDS	VSAMKSDS	VSAMLSDS	VSAMRRDS
Type	nonindexed	indexed	linear	numbered
Space (primary)	400KB	400KB	400KB	400KB
	Comments: Can be overridden by the SPACE parameter			
Space (secondary)	400KB	400KB	400KB	400KB
	Comments: Can be overridden by the SPACE parameter			
RECORDSIZE (avg)	256	256	256	256
	Comments: Is overridden by the LRECL parameter			
RECORDSIZE (max)	256	256	256	256
	Comments: Is overridden by the LRECL parameter			
BUFSPACE	45056	45568	8192	45056
	Comments: Can be overridden by the AMP parameter			
CISIZE	22528	22528	512	22528
	Comments: System determined optimum size for device (currently 3390). Can be influenced by RECORDSIZE (specified by the JCL LRECL parameter) and the BUFSP value (specified in the JCL DD AMP parameter)			
KEYS (length)	n/a	1	n/a	n/a
	Comments: Is overridden by the KEYLEN parameter			
KEYS (offset)	n/a	0	n/a	n/a
	Comments: Is overridden by the KEYOFF parameter			

ATTRIBUTE	VSAM DATA CLASSES			
	VSAMESDS	VSAMKSDS	VSAMLSDS	VSAMRRDS
SHARE- OPTIONS	1,3	1,3	1,3	1,3
	Comments: Can be modified with the IDCAMS ALTER command			
recovery/speed	RECOVERY	RECOVERY	RECOVERY	RECOVERY
erase/noerase	NOERASE	NOERASE	NOERASE	NOERASE
writechk/ nowritechk	NOWRITECHK	NOWRITECHK	NOWRITECHK	NOWRITECHK
	Comments: Can be modified with the IDCAMS ALTER command			
imbed/ noimbed	n/a	NOIMBED	n/a	n/a
	Comments: NOIMBED is recommended for systems with cache controllers such as those in use at NIH			
replicate/	n/a	NOREPLICATE	n/a	n/a
	Comments: NOREPLICATE is recommended for systems with cache controllers such as at NIH			
reuse/noreuse	NOREUSE	NOREUSE	NOREUSE	NOREUSE
spanned/ nonspanned	NONSPANNED	NONSPANNED	NONSPANNED	NONSPANNED
FREESPACE_ CI	n/a	20%	n/a	n/a
	Comments: Can be modified with the IDCAMS ALTER command			
FREESPACE_ CA	n/a	20%	n/a	n/a
	Comments: Can be modified with the IDCAMS ALTER command			

Figure 2-4 (cont.). Default Attributes for VSAM Data Classes (Continued)

NOTE: The space allocations are equivalent to (TRK,(10,10)) in terms of 3390 tracks.

Examples

The following examples illustrate how data classes can be used to allocate a new data set and how they can be modified to create a data set that does not exactly match the default values for the data class.

This example allocates a data set to hold the source for a COBOL program (up to 8000 records).

```
//DD1 DD DSN=aaaaiii.somename.COBOL,DATACLAS=DATAF,  
// UNIT=FILE,DISP=(NEW,CATLG)
```

This example creates a variable data set that could be used to save output from a job that has many short lines in its listing. Space may be saved by putting it into a variable file.

```
//DD1 DD DSN=aaaaiii.somename.LIST,DATACLAS=LISTV,  
// UNIT=FILE,DISP=(NEW,CATLG)
```

In this example, the listing has machine instead of ASCII control characters so the record format is overridden.

```
//DD1 DD DSN=aaaaiii.somename.LIST,DATACLAS=LISTV,  
// RECFM=VBM,UNIT=FILE,DISP=(NEW,CATLG)
```

This example has 90-character fixed length records. The LRECL parameter is used to override the default record length.

```
//DD1 DD DSN=aaaaiii.somename.DATA,DATACLAS=DATAF,  
// LRECL=90,UNIT=FILE,DISP=(NEW,CATLG)
```

NOTE: even though the LRECL has been increased from 80 to 90, the amount of space allocated will NOT be increased, i.e., the maximum number of 90-character records that will fit in this data set would be about 11% less than the number of 80-character records.

The following example illustrates overriding the default space values for the DATACLAS parameter.

```
//DD1 DD DSN=aaaaiii.somename.DATA,DATACLAS=DATAF,  
// LRECL=200,SPACE=(200,(50,5),RLSE),AVGREC=K,  
// UNIT=FILE,DISP=(NEW,CATLG)
```

The following examples illustrate the creation of VSAM data sets using the DATACLAS JCL parameter.

```
//DD1 DD DSN=aaaaiii.somename.KSDS,DATACLAS=VSAMKSDS,  
// LRECL=530,KEYLEN=45,KEYOFF=10,  
// UNIT=FILE,DISP=(NEW,CATLG)
```

```
//DD1 DD DSN=aaaaiii.somename.ESDS,DATACLAS=VSAMESDS,  
// LRECL=800,SPACE=(800,(100,10),RLSE),AVGREC=K,  
// UNIT=FILE,DISP=(NEW,CATLG)
```

2.5.17 Concatenating Data Sets

Concatenating data sets means logically connecting them for the duration of a job step. The two or more concatenated data sets will be read successively as a single data set. Sequential or partitioned data sets can be concatenated. For details on concatenating data sets, see the manual *OS/390 MVS JCL Reference*, GC28-1757.

Some considerations when concatenating data sets:

- To concatenate data sets, omit the ddnames from all DD statements except the first. The data sets are processed in the same sequence as the DD statements defining them.
- Concatenated data sets can reside on different device types only if the software being used permits this.

Example

```
//INPUT DD DSN=XYZ1ABC.FIRST.DATASET,DISP=SHR  
// DD DSN=XYZ1ABC.SECOND.DATASET,DISP=SHR
```

2.6 CREATING AND USING DATA SETS

Creating and accessing data sets is a primary concern for any computer application. Two general classifications of data sets are permanent and temporary. Permanent data sets can exist before and after a batch job executes. Temporary data sets can exist only for the duration of a single batch job. This section provides the requirements and restrictions for creating permanent and temporary data sets on tape and disk. The section also describes the techniques for referencing them in a job: via the system catalog, passed from a previous step, or directly (specifying all data set storage information).

2.6.1 Creating Temporary Data Sets on System Scratch Disks

Temporary data sets may be created and used on system scratch disks. Using scratch disks for temporary data storage is generally more efficient and less prone to problems arising from operator and hardware errors than using scratch tapes.

Required DD Parameters

- UNIT=SYSDA specifies that a system scratch disk is to be used.
- SPACE specifies the space to be allocated for the data set. See the *NIH Computer Center User's Guide* for details on estimating space for any new data set. Note that key length (KL) must be considered for some BDAM data sets when calculating space requirements on a 3390.

Optional DD Parameters

- DISP=(NEW,PASS) is needed if the data set is to be used in a later step.
- DSN can be used for ease of reference if a data set is passed; if no dsname is given, the system assigns a unique name. The dsname must be a temporary data set name.
- DCB is needed if the desired DCB characteristics for the data set are not supplied by the program.

Usage at NIH

- If a job exceeds the maximum of 100,000 tracks, it will be cancelled.
- It may not be possible to allocate the maximum of system scratch space to a single data set. The primary space must fit on a single disk. If the unit count field of the UNIT parameter on the DD statement is not coded, then all the secondary space must also be acquired on the same disk. If a single sequential, direct or partitioned data set requires a large space allocation, coding UNIT=(SYSDA,2) on the DD statement will enhance the chances of finding the needed space.

Examples

```
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,1),RLSE)

//TEMPOUT DD UNIT=SYSDA,SPACE=(11440,(310,10),RLSE),
// DISP=(NEW,PASS),DSN=&&OUT,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440)
```

2.6.2 Creating and Deleting Data Sets

The system catalog and catalog-maintenance routines provide the facility to allow the data set name to be identified with the volume serial number and unit type of the volume on which it is stored. Data sets may be entered into and deleted from the catalog by appropriate specification in the DISP parameter.

Required DD Parameters

- DSNNAME must be of the form aaaaiii.name.

Optional DD Parameters

- UNIT specifies the type of device to be used. Required when the data set is created.
- VOLUME specifies the volume information for the cataloged data set. Required only if the data set is being written to a tape or dedicated disk.
- LABEL must be specified for tapes if the data set is not the first data set on the tape.
- SPACE must be specified when creating disk data sets.

- DCB should be specified if the desired DCB characteristics for the data set are not supplied by the program or by the data set label.

Additional Information

- Only data sets (including tape data sets) having "standard" data set names can be cataloged. A standard data set name consists of two or more groups of one-to-eight characters, separated by a period. (The first group is called the high-level index.) The first character of each group must be alphabetic or national (#,@,\$); the remaining characters can be alphabetic, numeric, national or hyphen. The data set name, including the periods, must not exceed 44 characters in length.
- At NIH the high-level index must be a valid seven-character account/initials combination.
- Apostrophes are required for the data set names of non-standard data sets stored on tape. Non-standard data set names cannot be cataloged.
- If a cataloged data set is deleted (e.g., DISP=(OLD,DELETE)), but it was not referenced through the catalog (i.e., the DD statement specified the volume serial number and unit information for the data set), then the catalog entry for the data set will not be removed. Thus, the catalog will have an entry for the data set even though the data set no longer exists. In this case the DSSCR procedure should be used to remove the catalog entry.

Examples

```
//WORKFILE      DD DSN=ABC1XYZ.WORKDATA,UNIT=FILE,
//              DISP=(NEW,CATLG,DELETE),
//              SPACE=(TRK,(20,1),RLSE)

//STATFILE      DD DSN=ABC1XYZ.STATS,UNIT=MSS,
//              DISP=(NEW,CATLG,DELETE),
//              SPACE=(CYL,(5,1),RLSE),
//              DCB=(RECFM=FB,LRECL=150,BLKSIZE=11400)

//S1            EXEC PGM=IEFBR14
//ALLOCATE      DD DSN=aaaaiii.filename,
//              DISP=(NEW,CATLG),UNIT=FILE,
//              SPACE=(CYL,(1,1),RLSE)
```

2.6.3 Creating Permanent or Temporary Data Sets on Tape

Although temporary data sets may be created and used on system scratch tapes, they should not be unless the user has determined that they are required (e.g., because the permissible public scratch disk space has been fully used). Using scratch disks for temporary data storage is generally more efficient and less prone to problems arising from operator and hardware errors.

Required DD Parameters

- UNIT specifies the type of tape drive to be used for the data set. (See Section 2.5.12 for the valid device names.)

Optional DD Parameters

- DSN should be specified if the data set is a permanent data set. All permanent data set names should begin with valid account/initials. If the data set is temporary, DSN may be specified for ease of reference.
- DISP must be specified if the default of DISP=(NEW,DELETE) is not desired. DISP=(NEW,PASS) can be specified for either a temporary or permanent data set. DISP=(NEW,KEEP) can be specified only for permanent data sets.
- VOL must be coded if a specific tape volume is desired; otherwise, a scratch tape is used (and assigned to the user only if the data set is kept and the job is run Class B or C).
- LABEL should be supplied if the data set is not to be the first data set on the tape, if the tape is not an IBM standard labeled (SL) tape, or if the data set is to be read by a FORTRAN program. No more than 100 data sets should be put on one tape.
- DCB should be specified if the desired DCB characteristics for the data set are not supplied by the program.
- **Note: Even though this parameter is ignored on the South System, to prepare for the transition to Titan, code an expiration date.** Code either:
 - RETPD – retention period (the number of days; e.g., RETPD=5)or
 - EXPDT – expiration date (e.g., EXPDT=2003/180 for the 180th day of the year 2003)

Examples

```
//SORTOUT      DD UNIT=TAPE,DSN=&&XYZSRT,DISP=(NEW,PASS)

//GO.FT08F001  DD UNIT=TAPE,DSN=ABC1XYZ.SUMS,
//              LABEL=(2,,IN),DISP=OLD,
//              VOL=(PRIVATE,SER=123456),RETPD=5,
//              DCB=(RECFM=FB,LRECL=150,BLKSIZE=11400)
```

2.6.4 Using Permanent Data Sets

Permanent data sets can be stored on disk or tape. For information on the standards for each type of storage, see the *NIH Computer Center User's Guide*.

Required DD Parameters

- DSN specifies the data set name.
- DISP specifies the status of the data set both before and after it is referenced. The first DISP subparameter must be either OLD, MOD, or SHR.

Optional DD Parameters

- UNIT specifies the type of device assigned to a data set. Not needed for cataloged data sets.
- VOL specifies the volume the data set is stored on. Not needed for cataloged data sets.
- LABEL should be supplied if the data set is not the first data set on the tape, if the tape is not an IBM standard labeled (SL) tape, or if the data set is to be read by a FORTRAN program.
- DCB should be specified if the desired DCB characteristics for the data set are not supplied by the program or by a data set label. The DEN subparameter should be coded for 9-track tapes.

Examples

```
//TOTALS      DD DSN=ABC1XYZ.SUMS,UNIT=TAPE,
//              DISP=(OLD,KEEP),VOL=(PRIVATE,SER=123456),
//              LABEL=2

//SORTIN      DD DSN=ABC1XYZ.INDAT,DISP=SHR

//OUTPUT      DD DSN=ABC1XYZ.DATA,UNIT=FILE,
//              SPACE=(CYL,(15,2),RLSE),DISP=(NEW,CATLG)
```

2.6.5 Using a Passed Data Set

A passed data set is one which is allocated to a job step and is not deallocated at step termination. It remains available to subsequent steps of the same job.

Required DD Parameters

- DSN specifies the data set name of the passed data set.
- DISP specifies the status of the data set both before and after it is referenced. The first DISP subparameter must be OLD, MOD, or SHR.

Optional DD Parameters

- LABEL should be specified for tapes if the data set is not the first data set on the tape or if the tape is not an IBM standard label (SL) tape.
- DCB should be specified if the desired DCB characteristics for the data set are not supplied by the program or by the data set label.

Examples

```
//TEMPOUT      DD DSN=&&XYZOUT,DISP=(OLD,DELETE)

//S2.DATA      DD DSN=ABC1XYZ.DATA,DISP=(MOD,KEEP),
//              DCB=(RECFM=FB,LRECL=150,BLKSIZE=11400),
//              LABEL=(2,NL)
```

2.7 CATALOGED PROCEDURES

Production jobs that require many job control language statements can be simplified through the use of cataloged procedures. A cataloged procedure is a set of job control language statements that has been assigned a name and is placed in a partitioned data set known as a procedure library (JCLLIB). The procedure library is referenced by including a JCLLIB statement in the job. The cataloged procedure that contains the JCL statements is retrieved by giving its name on an EXEC statement in the input stream. Additions and modifications can be made with override statements (described in Section 2.7.6). For a more complete explanation of the uses and restrictions of cataloged procedures, see the manual *OS/390 MVS JCL Reference*, GC28-1757.

A procedure contains one or more procedure steps; each step consists of an EXEC statement that identifies the program to be executed and DD statements that define the data sets to be used or produced by the program. The program requested on the EXEC statement must exist in the system library or in a private library defined by a STEPLIB DD statement.

A cataloged procedure must not contain JOB statements, delimiter statements, null statements, JOBLIB DD statements, or DD statements with * or DATA coded in the parameter field.

WARNING: Users should avoid copying JCL from NIH Computer Center procedures whenever possible. The internals of procedures may be changed at any time without warning as long as there is no change in the way they appear to the user. Copied JCL may thus become unusable.

2.7.1 JCLLIB

Cataloged procedures are stored in partitioned data sets (PDSs) known as procedure libraries or JCL libraries. The JCLLIB facility is used in a job to specify libraries containing procedures and INCLUDE groups to be used in JCL. (See Section 2.7.8 for information on INCLUDE groups.) The format of the JCLLIB statement is:

```
//name JCLLIB ORDER=(aaaaiii.library1,bbbjjj.library2,...)
```

where name is optional and follows the rules for the ddname field in a DD statement. The JCLLIB statement must appear after the JOB statement and before the first EXEC statement. Each procedure library specified must be cataloged. The libraries are searched before any system procedure libraries and are searched in the order specified. For example,

```
//MYPROCS JCLLIB ORDER=(AAAAIII.MYLIB1,AAAACKK.MYLIB2)
```

specifies that the procedure libraries AAAAIII.MYLIB1 and AAAACKK.MYLIB2 will be searched for procedures in the JCL that follows it. If only one library is specified, the parentheses are optional.

2.7.2 Defining Symbolic Parameters in a Cataloged Procedure

Symbolic parameters allow a programmer using a cataloged procedure to modify the procedure easily for the duration of a job step. When programmers call the procedure, they use the EXEC statement to assign values to the symbolic parameters.

A symbolic parameter consists of one-to-seven alphanumeric and/or national (#, @, \$) characters preceded by a single ampersand. The first character must be alphabetic or national. In the cataloged procedure, specification of symbolic parameters for data set names, volume serial numbers, etc., permits the creation of a more general procedure. In this manner one procedure may suffice where several might otherwise be required.

In the following example of a cataloged procedure, the data set names and tape serial numbers for the input and output data sets and the region size are left as variables; i.e., they have no default values.

Examples

```
//UPDATE PROC CORE= ,  
// INPUT= ,  
// VOLI= ,  
// OUTPUT= ,  
// VOLO=  
//UPDATE1 EXEC PGM=WEEKLY,REGION=&CORE  
// INPUT DD DSN=&INPUT,UNIT=TAPE,VOL=SER=&VOL I ,  
// DISP=OLD  
//OUTPUT DD DSN=&OUTPUT,UNIT=TAPE,VOL=SER=&VOLO ,  
// DISP=( ,KEEP )
```

More detailed information on symbolic parameters can be found in the manual *OS/390 MVS JCL Reference*, GC28-1757.

2.7.3 Using Cataloged Procedures

To use a cataloged procedure, the procedure name is coded on the EXEC statement, and any necessary symbolic parameters are specified. To use the above example of a cataloged procedure, code:

```
//STEP1 EXEC UPDATE,CORE=200K,INPUT=dsname1,  
// VOLI=serial1,OUTPUT=dsname2,VOLO=serial2
```

Frequently it is necessary for the user to modify one or more parameters in a cataloged procedure in order to make use of it. A procedure can be modified by assigning values to or nullifying symbolic parameters contained in the procedure; by overriding, adding, or nullifying parameters on EXEC statements in the procedure; or by adding or overriding DD statements in the procedure.

2.7.4 Assigning Values to and Nullifying Symbolic Parameters

Many procedures contain symbolic parameters which must be assigned values or nullified unless the default values are satisfactory. For example, to use the NIH Computer Center procedure DSLIST to list the third data set, ABC1XYZ.DATA on a 9-track, 6250 BPI tape:

```
// EXEC DSLIST,NAME='ABC1XYZ.DATA',STORAGE=9TRACKHI,SEQ=3
```

Symbolic parameters need not be coded in any specific order when they are assigned values or nullified.

2.7.5 Modifying Parameters on the EXEC Statement

A cataloged procedure must contain one or more EXEC statements, one for each program to be executed. Each EXEC statement can contain keyword parameters to allow conditional execution of the step, to specify a region size for the step, or to pass information to the program executed in the step. These EXEC statement parameters are COND, REGION, and PARM. For an explanation of the meaning and use of these parameters, see Section 2.4.

To add one of these parameters to a step in a procedure, or to override a parameter that already exists, the user must code:

```
parameter.procstepname=value
```

on the EXEC statement invoking the procedure. "parameter" is one of the keyword parameters listed above, and "procstepname" is the name of the step in the procedure to which the parameter will apply. For a multi-step procedure, all overrides or additions to one step must be coded before the overrides or additions to a succeeding step.

The following EXEC statements would be valid for a three-step procedure named IIIPROC with stepnames GO1, SORT2, and PRINT3:

```
//S1 EXEC IIIPROC,REGION.GO1=700K,COND.SORT2=(16,EQ),  
// PARM.PRINT3=NUMBER  
  
//S1 EXEC IIIPROC,REGION.GO1=1200K,REGION.SORT2=300K,  
// PARM.SORT2='SIZE=204800',COND.PRINT3=(0,NE,SORT2)
```

The following EXEC statement is invalid because the parameters are not in procedure step order:

```
//S1 EXEC IIPROC,REGION.SORT2=400K,REGION.GO1=660K,  
// COND.PRINT3=(4,EQ),COND.GO1=(4,NE)
```

If an EXEC statement parameter is coded without a qualifying stepname, it will apply to the entire procedure. The parameter PARM, when used without a stepname, will apply to the first step and nullify all succeeding PARM fields. COND applies to every step in the procedure and overrides any COND information given in the procedure. REGION applies to every step and overrides all REGION values coded in the procedure.

2.7.6 Overriding or Adding DD Statements

It may be necessary to override or add DD statements to a procedure. These DD statements may: define data card images in the input stream, define a data set for an ABEND dump, describe a data set on tape or disk, define a printer, or override any DD statements or parameters on DD statements already in the procedure.

An overriding or added DD statement must have the form:

```
//procstepname.ddname DD parameters
```

where "procstepname" identifies the procedure step the user is overriding and "ddname" is the DD statement in that procedure step. The following DD statement is an example:

```
//GO.MASTIN DD UNIT=TAPE
```

"MASTIN" is the name of the DD statement and "GO." refers to the name of the procedure step to which this override or addition applies.

If the user adds a ddname, it must be different from any ddname already in the procedure step. For a multi-step procedure, all overrides or additions to one step must be coded before the overrides or additions to a succeeding step.

The overriding DD statement contains only those keyword parameters that are to be replaced, added or nullified. To replace or change part of a DD statement parameter, the user must code the entire parameter with its new value on the overriding DD statement. The parameter values in the procedure are replaced by the values assigned to those parameters on the override DD statement. (The DCB parameter is the only exception to this rule, i.e., it is possible to add or replace subparameters in the DCB without causing the entire field to be replaced. DCB=BLKSIZE=11438 on an override statement will merge this blocksize with the other DCB characteristics already coded for the DD statement in the procedure.)

To add or replace parameters, code them on the override DD statement in any order. To nullify parameters, code the keyword followed by an equal sign. In the DD statement below, both the UNIT and SPACE parameters are nullified on the OUTPUT DD statement in the GO step.

```
//GO.OUTPUT DD UNIT=,SPACE=
```

The DCB parameter is nullified only when each subparameter is nullified.

```
//GO. OUTFILE DD DCB=(RECFM=,LRECL=,BLKSIZE=)
```

Certain parameters are mutually exclusive and coding one of them on an override statement will cause the other to be nullified. Do not code the nullified parameter. DISP and SYSOUT, DSNNAME and DUMMY are examples of mutually exclusive parameters.

Procedure DD statements, which have a SPACE parameter with RLSE or an incremental quantity coded, should have the SPACE parameter nullified when the override DD statement defines an existing disk data set. If the SPACE parameter is not nullified, the incremental quantity will be applied to the data set, and any unused space in the existing data set will be released.

Override DD statements which have * or DATA coded in the operand field completely negate any parameters coded in the procedure's DD statement.

To be sure of the technical correctness of overrides to procedures maintained by the NIH Computer Center, use the JCLSCAN utility (Section 3.12) before running the job. This will provide a listing of the procedure and the accompanying override statements without actually running the job. The NIH Computer Center reserves the right to change, without warning, any of its procedures at any time if the change will not affect published usage of the procedure.

2.7.7 Instream Procedures

An instream procedure is a set of job control language statements in the form of card images that have been placed in the input stream of a job. Instream procedures can be extremely useful in testing procedures that are to be placed in a procedure library and in setting up a set of JCL for repeated use during a single job. An instream procedure can be executed any number of times during the job in which it appears, and up to fifteen uniquely named instream procedures can appear in one job. An instream procedure consists of one or more steps called procedure steps. Each procedure step consists of an EXEC statement and DD statements.

To use an instream procedure, submit a JOB statement followed by the instream procedure to be used. It can appear immediately following the JOB statement or the JOBLIB DD statement. The instream procedure cannot appear before the JOB statement or after the EXEC statement that invokes it.

An instream procedure is subject to the same restrictions as a regular procedure. It must not contain JOB statements, delimiter statements, null statements, JOBLIB DD statements, or DD statements with * or DATA coded in the operand field.

To use an instream procedure, the procedure name is coded on the EXEC statement, and any necessary symbolic parameters are specified. The user may override parameters, modify DD statements contained in the procedure, and add DD statements to the procedure.

An instream procedure must be preceded by a "PROC" statement and followed by a "PEND" statement.

```
//BACKUP      PROC  DSNNAME= ,
//            STORAGE=FILE ,
//            TAPE= ,
//            UNIT=TAPE
//STEP1       EXEC  PGM=COPYIT,REGION=500K
//DD1         DD   DSN=&DSNAME,DISP=SHR,UNIT=&STORAGE
//OUT         DD   DSN=&DSNAME,DISP=(NEW,KEEP) ,
//            UNIT=&UNIT,VOL=SER=&TAPE,DCB=DEN=&DEN
//STEP2       EXEC  PGM=SCRATCH,COND=(0,NE)
//DEL         DD   DSN=&DSNAME,DISP=(OLD,DELETE) ,
//            UNIT=&STORAGE
//            PEND
```

More detailed information on symbolic parameters can be found in the manual *OS/390 MVS JCL Reference*, GC28-1757.

The "operands" field of the PROC statement either contains values for symbolic parameters or is blank. The "name" and "comments" fields on the PEND statement are optional.

The above procedure may be placed within the job stream and may be invoked any number of times. The following example illustrates how the user might override and add to the DD statements of the instream procedure listed above:

Examples

```
//BEGIN       EXEC  BACKUP,DSNAME='AAAAIII.OLDMAST' ,
//            TAPE=123456
//NEXT        EXEC  BACKUP,DSNAME='AAAAIII.OLDDATA' ,
//            STORAGE=TMP,TAPE=123456
//STEP1.OUT   DD   LABEL=(2,SL)
//STEP2.DEL   DD   DISP=(OLD,KEEP)
```

2.7.8 INCLUDE Groups

An INCLUDE group is a set of JCL statements that may be included in a JCL job stream. The JCL statements must be saved as a member of a cataloged procedure library that is specified on a JCLLIB statement. For more information on JCLLIB, see Section 2.7.1. An INCLUDE group may not contain any of the following:

- JOB statements
- PROC and PEND statements

- JCLLIB statements
- JES2 control statements (statements beginning with a /*)
- DD * and DD DATA statements.

For example, if the following statements are saved as member PRINT in a PDS named AAAAIII.MYLIB:

```
/* * PRINT JCL
//PRINT EXEC COPY
//SYSUT1 DD DSN=&OLDDATA,DISP=(OLD,DELETE)
//SYSUT2 DD SYSOUT=A
```

Then the user submits the following job stream using the WYLBUR RUN command or the TSO SUBMIT command:

```
//III JOB (AAAA,999,E),USERNAME
/* *
//LIB JCLLIB ORDER=AAAAIII.MYLIB
/* *
//STEP1 EXEC PGM=MAIN
:
:
//OUTPUT DD DSN=&OLDDATA,DISP=(NEW,PASS)
/* *
// INCLUDE MEMBER=PRINT
```

Any output written by program MAIN to the DDname OUTPUT will be passed (via the temporary data set name &OLDDATA) to the next step (which was INCLUDED) and printed on the central printers.

2.8 JOB DEPENDENCY CONTROL STATEMENTS

In a highly complex facility with a large number of users, individuals sometimes need to be able to control the order in which jobs execute and to prevent more than one of a group of jobs from executing at the same time. The control statements in this section give users these abilities. Only one of these control statements should be included in a job; if there are more, all but the last will be ignored. Because use of these control statements prevents the system from selecting jobs in their natural sequence, job turnaround times may be increased.

The cross-system enqueue software in use at the NIH Computer Center provides automatic data set integrity protection in a multi-system environment, preventing accidental destruction of data from jobs or online sessions modifying any data set with the same name at the same time. For further information on cross-system enqueue see the *NIH Computer Center User's Guide*. Jobs that require access to a data set that is being updated will be suspended until the controlling job or session relinquishes control. Because the suspension of job execution can have a negative impact on the performance of the system, any job that has been suspended for more than 30 minutes will be cancelled. This problem can be avoided by using the

`/*CNTL`, `/*AFTER`, and `/*BEFORE` facilities which hold the potentially conflicting jobs in the job queue rather than suspending job execution after it has already begun.

The `/*ROUTE XEQ LATE` statement (Section 2.10.10) offers another technique for controlling job execution. Any job containing this statement will start execution between 10:00 p.m. and 6:00 a.m.

The DSSUBMIT program (Section 3.11) allows precise control over execution of a series of jobs.

2.8.1 `/*AFTER` and `/*BEFORE`

`/*AFTER` and `/*BEFORE` permit the user to control the order in which two or more jobs will be executed while making certain that only one executes at a time. Jobs must be submitted in the order they are to be executed to insure the desired result.

```
/*AFTER jobname
```

delays the running of the job which contains this statement until after the job (if there is one with the specified jobname) has completed execution.

```
/*BEFORE jobname
```

causes the job which contains it to complete execution before the job with the specified job name begins execution.

The WYLBUR commands `SHOW JOBS AFTER jobname` and `SHOW JOBS BEFORE jobname` will list all jobs executing or awaiting execution that contain a `/*AFTER` or `/*BEFORE` statement with a "jobname" beginning with the specified jobname.

2.8.2 `/*CNTL`

The `/*CNTL` facility permits users to designate groups of jobs which can be prevented from running when specifically chosen jobs run.

There are two `/*CNTL` statements which may be included in a job:

```
/*CNTL    controlname,SHR
/*CNTL    controlname,EXC
```

where "controlname" is a maximum of 8 characters beginning with a set of registered initials. SHR is the default. There can be no embedded blanks or commas.

EXC `/*CNTL controlname,EXC` insures that no other job with the same controlname will be executed simultaneously.

SHR `/*CNTL controlname,SHR` insures that no job which contains a `/*CNTL`

controlname,EXC statement with the same controlname will be executed simultaneously.

The /*CNTL statements are typically used to regulate the execution of jobs which access a private resource shared by many users. The resource may be a tape containing one or more data sets or a disk data set. A /*CNTL statement with the same controlname is included in each job which accesses the resource. Jobs which may share the resource (e.g., reading a program library) contain a /*CNTL controlname,SHR statement. Jobs which must have exclusive use of the resource (e.g., updating a program library or using a tape) contain a /*CNTL controlname,EXC statement.

The owner of the resource should maintain contact (e.g., via a mailing list) with the individuals who access the data to coordinate data set updates (and other accesses requiring exclusive control) and to designate the controlname. The user whose initials are used in the controlname can determine who uses the controlname. If a user reports to TASC that jobs using the controlname without permission are causing delays, those users will be contacted and their jobs cancelled.

If multiple jobs with the same /*CNTL EXC statement are awaiting execution at the same time, they will be executed in strict job class order (i.e., Class A jobs will execute before Class B). If a Class A job is set to level 0 (so it will be held and not run), all /*CNTL EXC Class E, B and C jobs with that controlname will also be held.

The WYLBUR command SHOW JOBS CONTROL controlname will list all jobs executing or awaiting execution that have a "controlname" beginning with the specified controlname.

Since /*AFTER and /*BEFORE should not be used in combination with /*CNTL, the user may wish to use DSSUBMIT to control the order of execution. (DSSUBMIT is described in Section 3.11.)

While jobs executed at the NIH Computer Center are expected to execute within the time limits set for their job classes (given in Section 2.2.4), there is not normally a strict order of job execution by classes. (For example, Class E jobs may execute before Class A jobs.) However, when multiple jobs with the same controlname are awaiting execution at the same time, their order of execution will follow certain rules if at least one of the jobs contains a /*CNTL controlname,EXC statement.

These rules are:

- Any jobs containing EXC statements will execute in strict job class order. The job class order is:
 - F
 - A
 - E
 - B

For example, if a Class A and a Class E job are awaiting execution and each has the same /*CNTL controlname,EXC statement, the Class A job will execute first.

- SHR jobs will execute before EXC jobs lower in the job class order and after EXC jobs higher in the job class order.

Let's look at an example where Class A, E, and B jobs are awaiting execution and the Class B job contains a /*CNTL XYZ,EXC statement while the A and E jobs contain /*CNTL XYZ,SHR statements. First, the A and E jobs will complete execution. (However, the E job could execute before, after, or at the same time as the A job.) Then, the B job will execute (after both the A and E jobs have finished).

- If a job that is higher in the job class order than an EXC job is set to priority level 0, the EXC job will not execute until the priority level is raised and the higher job runs. Similarly, a job delayed while waiting for a resource (RECOVERY, etc.), can delay other jobs.
- Requesting DISCOUNT for one or more of the jobs will not cause others with the same controlname to be delayed.

2.9 RESOURCE CONTROL STATEMENTS

The /* statements in this section are used in jobs that access resources that may not be available on all processor complexes.

2.9.1 /*MESSAGE

If a job requires the mounting of any pre-assigned tapes, it must include the following "message" statement immediately following the JOB statement:

```
/*MESSAGE xxxxxx,v;yyyyyy,v
```

where "xxxxxx" and "yyyyyy" are the volume serial numbers of the tapes requested; v may be replaced with an R to allow reading only or a W to permit reading or writing; R is the default. There must be at least one blank after the word MESSAGE. The message portion of the message statement cannot extend beyond column 71. Although message statements may not be continued, as many as are necessary may be used. Up to 196 tapes can be requested in the /*MESSAGE statements for a job. If a private tape is requested by a job and no message statement has been used to convey this information, the job will ABEND. No message statement is needed for disks.

Users who are processing tapes that are not part of NIH Computer Center's tape library, (for example, tapes acquired from other installations or tapes used to record laboratory data), should handle these tapes as follows:

- Give the tape to Output Distribution Services before the job is submitted.

- Include the following message statement in the job:

```
/*MESSAGE volser,vS
```

where "volser" is the exact volume serial number.

This tells the operators that the tape called for by that job is a special user tape and will come from Output Distribution Services rather than the tape library. For further information, the *NIH Computer Center User's Guide*.

User comments may be included by leaving at least one space after the last volume element and enclosing the comment within single quotes. This facility is for user documentation only and may not be used for operator communication.

Examples

```
/*MESSAGE MYTAPE,RS
```

```
/*MESSAGE 010560,W 'OUTPUT TAPE FOR STEP1'
```

```
/*MESSAGE U00921,R;000321,R;U00084,W
```

2.9.2 /*ROUTE XEQ

The /*ROUTE XEQ mechanism allows a user to request specific routing for a job (e.g., execution on a processor complex containing a resource not simultaneously available on all the subsystems or to request execution during certain hours).

```
/*ROUTE XEQ resource
```

where "resource" is a type of hardware or a facility.

Any job which processes tape, accesses DB2, or uses other routable resources must include an appropriate /*ROUTE XEQ resource statement to insure that it runs on a subsystem with the needed resource. This is especially true for jobs that might run during time periods in which not all services or resources are available (such as a holiday unattended service period).

The following is a summary of the available resources and their uses:

Resource	Function
DB2	to run DB2 or QMF in the batch
TAPE	IBM standard tape
CTAPE	3480 cartridge "special" tape
ETAPE	3490E enhanced tape
9TRACKHI	9-track 6250 BPI or 1600 BPI tape

FICHE	microfiche tape
RECOVERY	to restore a dedicated (private) disk from a backup tape
LATE	to delay execution to begin between 10:00 p.m. and 6:00 a.m. If a job with LATE specified is not started before 6:00 a.m., it will be held until the next LATE period.
VTAM1	route a job to the system running CONNECT:Direct.

2.9.3 /*ASSIGN

To assign scratch tapes kept by a Class B or Class C job to a set of initials other than those on the JOB statement, include a

```
/*ASSIGN aaaaiii.kkk
```

statement in the job where "aaaa" is the account and "iii" is the set of initials to which all tapes assigned by the job should be assigned. "kkk" is the keyword for that aaaa iii combination. A "/" may be substituted for the "." between the initials and keyword. If the account or initials specified are invalid or the keyword is incorrect, the statement will be ignored. If more than one /*ASSIGN statement is included, the last one will be used.

2.9.4 /*ACCESS

A /*ACCESS control statement must be included in any job which includes a /*MESSAGE statement for a non-public NIH Computer Center tape assigned to an account or initials different from those on the JOB statement. A /*ACCESS control statement cannot extend beyond column 71.

```
/*ACCESS aaaaiii
```

where "aaaaiii" is the account/initials combination to which the tape is assigned.

If non-public tapes with more than one owner are to be accessed, the account/initials combinations may be included in multiple /*ACCESS statements or listed on the same statement separated by semicolons.

```
/*ACCESS aaaaiii;aaa2iii;aaa2jjj
```

If a /*ASSIGN statement is included for a particular account/initials combination, they need not be respecified in the same job on a /*ACCESS statement.

/*ACCESS statements are not needed to access tapes with a "public" status. The VOLSTAT procedure can be used to change the status of a tape to PUBLIC from the NOPUBLIC default; see Section 3.5.6 for details.

2.10 JOB EXECUTION CONTROL STATEMENTS

The /*KEYWORD, /*PASSWORD, /*NOTIFY, /*DISCOUNT, /*QUICK, /*PRIORITY and /*RERUN statements are covered in this section.

2.10.1 /*KEYWORD

The keyword facility will be eliminated in the near future. All disk data set protection should be controlled by RACF. See the *NIH Computer Center User's Guide*.

The keyword facility protects the user's account and initials against unauthorized use in batch jobs. The following statement must be included or the job will not run. It need not be supplied for jobs submitted through WYLBUR or TSO since both automatically supply it.

```
/*KEYWORD=kkk
```

where "kkk" is the keyword.

When the JCL for the job is printed on the job output, the keyword will be replaced with blanks. Additional information on keywords, is given in the *WYLBUR Fundamentals* manual and the *NIH Computer Center User's Guide*.

2.10.2 /*NOTIFY

The NOTIFY facility allows users who are signed on to WYLBUR or TSO to be notified when their batch jobs reach completion. Inserting a

```
/*NOTIFY
```

statement into any batch job causes a message to be sent to both TSO and WYLBUR when it completes execution and again when its output is completed. The message is sent to the initials on the job statement provided the user is logged on to WYLBUR or TSO at the time; if the user is not logged on, the message is not sent.

```
/*NOTIFY iii
```

can be used to send the message to the specified initials. Only one user can be notified; so, if this option is used, the messages will not be sent to the initials on the JOB statement.

2.10.3 /*DISCOUNT

The DISCOUNT service allows users to receive a 50 percent rate reduction on batch processing for all jobs starting between 5:00 p.m. and 7:00 a.m. weekdays and all day weekends. To request the DISCOUNT facility include a

`/*DISCOUNT`

statement after the JOB statement along with any other JES2 control statements needed or include DISCOUNT as an operand in a WYLBUR RUN or RETRIEVE command or a TSO LISTOFF or SUBMIT command. No discount is offered on offline output services such as microfiche.

Any job containing a `/*DISCOUNT` statement will automatically be held for execution until the next discount period begins. For example, if a job which specifies DISCOUNT is submitted during the day (non-discount period), it will be held and automatically released to the system at 5:00 p.m. If a job with DISCOUNT specified is not started by 7:00 a.m. the following day, it will be held until the next evening, unless NODISCOUNT is specified, through WYLBUR or with the RJE workstation command described in the *Remote Job Entry Workstation Guide*.

Even though a discount job will be executed only during the discount period, all special tapes must be at the NIH Computer Center when the job is submitted.

Even if a job does not include a `/*DISCOUNT` statement, any job which begins execution during the discount period will receive the discount rate.

2.10.4 /*QUICK

The QUICK turnaround facility was established to enable a user who is waiting for output during the evening hours to bypass the long nighttime production runs. It is designed to help users who are actually present and cannot be used via the batch mode of WYLBUR. The facility is activated by the inclusion of a

`/*QUICK`

statement in the job or by specifying QUICK as an operand on the RUN, LIST, and RETRIEVE commands (WYLBUR) or the SUBMIT and LISTOFF commands (TSO). QUICK can be specified only on jobs being submitted during the 5:00 p.m. to 7:00 a.m. time period. If QUICK is specified at any other time or from the batch version of WYLBUR, a message will be sent, and the job will be run as a regular (non-QUICK) job of its class. QUICK should not be used for a priority level 0 job.

2.10.5 /*PRIORITY

Normally the priority of a job is the maximum for its job class. The priority level controls the order in which jobs are selected for processing from a given class. The maximum levels for each class are A-9, B-8, C-7, E-8, and F-9. A lower initial level may be set by including a

`/*PRIORITY n`

statement immediately preceding the JOB statement or by including priority on WYLBUR's RUN or LIST commands. If the user changes the level of a job in the queue, the job is

removed from the queue and re-added with the new priority; it is put at the end of the queue for the specified class and priority. Priority level 0 has a special meaning: a job with priority 0 will not be run until the priority is increased.

2.10.6 /*RERUN

If a job should be automatically resubmitted in the event of a system crash, the following statement should be included:

```
/*RERUN
```

/*RERUN should not be used if data sets on permanent volumes are created or modified during a job because status after a system failure is unpredictable. The RERUN parameter of WYLBUR's RUN command has the same function.

2.10.7 /*PASSWORD

To access a RACF-protected data set with other than universal access, the following statement must be included or the job will not run. It may be omitted for jobs submitted through TSO or WYLBUR since both will prompt for the password provided the USER parameter is supplied for the JOB statement.

```
/*PASSWORD=password
```

where "password" is the four-to-eight character password composed of alphanumeric or national (@,\$,#) characters.

Users who submit job control language that is stored in online data sets should use project initials instead of user initials. The passwords associated with project initials are not subject to expiration limits.

When the JCL for the job is printed, the password will be replaced with blanks. For further information on RACF protection for data sets, see the *NIH Computer Center User's Guide*.

2.10.8 /*XMIT

The /*XMIT statement is used to transmit records to a remote host. The remote host must be defined to the NIH system as a JES2 NJE node. JES2 builds header and trailer records from information on the JOB statement immediately preceding the /*XMIT statement and then transmits the records following the /*XMIT statement. If the records to be transmitted contain /*, a delimiter (DLM) must be specified on the /*XMIT statement to indicate that all records up to the delimiter are to be sent. The format of the /*XMIT statement is

```
/*XMIT nodename DLM=dd
```

where "nodename" is the name of a node or other node defined to NIH's JES2 with which files are to be transferred. The remote host must be an OS/390 JES2 or JES3 node, a VM

RSCS node, or a VSE POWER node. "dd" is a delimiter that indicates the end of the JCL records that are to be transmitted to the remote host.

/*XMIT is used in conjunction with the SENDFILE and RCVFILE programs. In order to use /*XMIT to send or receive files from a remote host, the remote host must support SENDFILE and RCVFILE. For more information on host-to-host file transfer, see Section 3.16.

2.10.9 /*NETACCT

The /*NETACCT statement specifies the accounting information required for jobs sending output to other computer nodes (such as NIHJES2 for the OS/390 Titan system of the NIH Computer Center) via the NJE (Network Job Entry) facility. The /*NETACCT statement can be included in the job by using the NETACCT option of WYLBUR's LIST OFFLINE, RUN, and SET BATCH commands. The format of the /*NETACCT statement is

```
/*NETACCT abcde
```

where "abcde" is the accounting information at the remote node. The accounting information can use up to 8 alphanumeric characters. The exact format of the accounting information varies according to the coding requirements of the receiving computer node.

2.10.10 /*ROUTE XEQ LATE

The /*ROUTE XEQ LATE statement is used to start executing the job between the hours of 10:00 p.m. and 6:00 a.m. The format of the /*ROUTE XEQ LATE statement is

```
/*ROUTE XEQ LATE
```

2.11 OUTPUT CONTROL STATEMENTS

The /*ROUTE, /*JOBOUT, /*DDOUT, OUTPUT and /*NOPURGE statements included in this section allow the user to direct output to desired locations and define its handling.

2.11.1 /*ROUTE

The /*ROUTE statement provides for routing output to an RJE workstation, the central computer facility, or the OUTPUT HOLD queue. If conflicting /*ROUTE statements referring to the same output are included in a job, the last one given will be used. The format of this statement is:

```
/*ROUTE type destination
```

where "type" can be PRINT or OUTPUT (for print), and "destination" can be:

Rn	to produce output at a remote batch workstation or VTAM printer.
----	--

CENTRAL or LOCAL	to produce output at the central computing or LOCAL facility.
HOLD	to direct printed output to the OUTPUT HOLD queue where it can be accessed.
HOLD,NIHCU	to direct printed output of a job that has been sent from another node to remain in the NIH OUTPUT HOLD queue after execution.
print command name [*]	to produce output at an VPS-defined printer.

Jobs awaiting output (print or punch) for more than two weeks will be purged.

Output which is left in the OUTPUT HOLD queue will be purged after 24 hours unless it is accessed (in which case it will be purged 4 hours after being accessed). The hours from midnight Friday to midnight Sunday are not counted as part of the 24-hour period a job has been in OUTPUT HOLD, but weekday holidays do count. If a job is purged from OUTPUT HOLD, no printed output is created. Output may be printed at the end of the waiting period instead of being purged; see Section 2.11.5.

A job must be released from the OUTPUT HOLD queue before any output data sets created by that job can be printed. (This would include output done at other destinations or on special forms.)

SYSOUT data sets may be printed at locations other than that specified on the ROUTE statement through use of the DEST parameter on a /*DDOUT or SYSOUT DD statement.

In case of systems problems, jobs in OUTPUT HOLD may be printed at the central facility. This does not occur frequently, but if it becomes necessary, no refund will be made even though the output of a job may no longer be available at a remote location.

Examples*

```
/*ROUTE PRINT "PRINT COMMAND NAME"

/*ROUTE OUTPUT CENTRAL
```

2.11.2 /*JOBOUT

The /*JOBOUT statement allows the default values for many output characteristics for the job to be overridden. A /*JOBOUT statement cannot extend beyond column 71. A /*JOBOUT statement cannot be continued, but more than one may be coded. If a job

^{*} For information on print command names for the mainframe-to-network printing system, refer to <http://silk.nih.gov/silk/vps>.

contains multiple /*JOBOUT statements with different operands, they will all take effect; if an operand is repeated, the last one takes precedence. Values for forms and copies on a SYSOUT statement will override those for a single data set, and can be overridden by those on a /*DDOUT statement.

The format of this statement is:

```
/*JOBOUT  FORMS=form,CHARS=( ttnn,ttnn ),COPIES=c,BOX=b,  
/*JOBOUT  DENSITY=d,PAGE=ln,FCB=fcb,EJECT=x,IMPACT=x
```

where

"form"	specifies the paper to be used for output.
"ttnn"	specifies the character sets (fonts) to be used. Parentheses are not needed when only one font is specified.
"c"	specifies the number of copies to be printed. The maximum is 250; the default is 1.
"b"	gives the computer output box to receive the output (e.g., 999 or M999).
"d"	specifies the vertical spacing in lines per inch. Laser printers permit 6, 8, and 10 lines per inch. Impact printers permit 6 and 8 lines per inch.
"ln"	specifies the number of lines on a page before an automatic eject is to occur. An integer value of up to 255 may be given. The default is to fill all but a half inch margin at the top and bottom of the page (e.g., 60 lines at 8 lines-per-inch on 8-1/2 inch paper).
"fcb"	specifies the four-character forms control buffer to be used to control vertical spacing for the output. Each form has a default FCB.
"x"	Y for yes or N for no. The default for IMPACT is N. The default for EJECT is Y; N causes printing to be done continuously without breaks for pages.

Complete details on the capabilities of the laser and impact printers are described in the *NIH Computer Center User's Guide*. The *User's Guide* includes information on forms, FCB's, character sets, and the interactions between the various printing options. SYSOUT DD statements for a single data set are described in Section 2.5.2.

Impact printing is required for output that cannot be produced on the faster laser printers. See the *NIH Computer Center User's Guide* for details.

Example

```
/*JOBOUT  FORMS=L2S1,COPIES=200
```

2.11.3 /*DDOUT

The /*DDOUT statement allows specification of output requirements for a group of SYSOUT data sets. Values on a /*DDOUT statement override any given in related SYSOUT statements or a /*JOBOUT statement. A /*DDOUT statement cannot extend beyond column 71, but it may be continued with an "*" in the code field. If two statements are given with the same code, only the first will be used. However, if an operand is repeated within a statement, the last one takes precedence. The format of this statement is:

```
/*DDOUT code FORMS=form,CHARS=(ttnn,ttnn),COPIES=c,  
/*DDOUT * DEST=(d1,d2,d3,d4),BOX=b,DENSITY=d,  
/*DDOUT * PAGE=ln,FCB=fcb,  
/*DDOUT * EJECT=x,IMPACT=x
```

where

"code"	is a 1-to-4 character (alphanumeric or national) name used to relate individual SYSOUT DD statements to this /*DDOUT statement. This code is specified on the SYSOUT DD statement as: //ddname DD SYSOUT=(A, ,code)
"form"	specifies the form to be used for job output.
"ttnn"	specifies the character sets (fonts) to be used. Parentheses are not needed when only one font is specified.
"c"	specifies the number of copies to be printed. An integer value of up to 250 may be given. The default value is 1.
"d1,d2,d3,d4"	specifies output destinations for the SYSOUT statements related to this /*DDOUT statement. A maximum of 4 values may be given to specify remote workstations (e.g., R8), VTAM printer, or the central facility (CENTRAL). If only one destination is given, the parentheses and separating commas are not required.
"b"	specifies the computer output box to receive the output (e.g., 999 or M999).
"d".	specifies the vertical spacing in lines per inch. Laser printers permit 6, 8, and 10 lines per inch. Impact printers permit 6 and 8 lines per inch.
"ln"	specifies the number of lines on a page before an automatic eject is to occur. An integer value of up to 255 may be given. The default is to fill all but a half inch margin at the top and bottom of the page (e.g., 60 lines at 8 lines-per-inch on 8-1/2 inch paper).

"fcb"	specifies the four-character forms control buffer to be used to control vertical spacing for the output. Each form has a default FCB.
"x"	Y for yes or N for no. The default for IMPACT is N. The default for EJECT is Y; N causes printing to be done continuously without breaks for pages.

Complete details on the capabilities of the laser and impact printers, the forms, FCBs, and character sets available, and the interactions between the various printing options are given in the *NIH Computer Center User's Guide*. The SYSOUT DD statements for a single data set are described in Section 2.5.2.

Examples

To cause most of a job's output to be printed using the system defaults but to have two SYSOUT data sets printed with 55 lines per page on 999L paper using the Orator Bold 10-pitch (OB10) font (character set):

```
/*DDOUT XYZ FORMS=999L,CHARS=OB10,PAGE=55
.
.
//DDN1 DD SYSOUT=(A,,XYZ)
.
.
//DDN6 DD SYSOUT=(A,,XYZ)
.
.
```

To cause most of a job's output to be sent to box 999 as specified on the JOB statement, while directing the SYSOUT data set, REPORT, to be mailed to the address for box M200.

```
//IIIRUN JOB (AAAA,999),NAME
/*DDOUT ABC BOX=M200
.
.
.
//REPORT DD SYSOUT=(A,,ABC)
```

To cause four copies of all except two SYSOUT data sets to be printed at Remote 8. For these two data sets (REPORT1 and REPORT5), three copies will be printed on Remote 8 and three more will be printed at the central printers. This job will probably need to have a value for lines printed specified on the JOB statement to allow enough lines for the multiple copies.

```
/*ROUTE PRINT R8
/*JOBOUT COPIES=4
/*DDOUT ABC COPIES=3,DEST=(R8,CENTRAL)
.
.
//REPORT1 DD SYSOUT=(A,,ABC)
```

```

      .
      .
//REPORT5 DD SYSOUT=(A, ,ABC)
      .

```

2.11.4 OUTPUT JCL Statement

The OUTPUT JCL statement is used only to request single- or double-sided printing on cut-sheet paper using the cut-sheet laser printer. With the OUTPUT statement, individual SYSOUT data sets or the whole job can be directed to the cut-sheet printer. For any job output data set referenced by an OUTPUT statement, the OUTPUT statement overrides /*JOBOUT values. /*DDOUT statements should not be used for SYSOUT data sets that reference OUTPUT statements; all specifications for the data set should be contained in the OUTPUT statement. The complete format of the OUTPUT statement is:

```

//name OUTPUT FORMDEF=fdef,FORMS=form,FCB=fcb,
//      CHARS=cccc,COPIES=nnn,DEFAULT=YES,
//      JESDS=ALL,LINECT=tt,TRC=YES

```

where

"name"	specifies a name used to relate individual SYSOUT DD statements to this OUTPUT statement. This name is specified on the SYSOUT DD statement as: //ddname DD SYSOUT=A,OUTPUT=*.name
"fdef"	specifies the FORMDEF form definition, which indicates whether single-or double-sided printing is desired.
"form"	specifies a form to be used for output (999P, 999L, 900P, or 900L).
"fcb"	specifies the four-character forms control buffer to be used to control vertical spacing of the output. Each form has a default FCB.
"cccc"	specifies the character set (font) to be used.
"nnn"	specifies the number of copies to be printed (up to 250). The default value is 1.
DEFAULT=YES	forces all SYSOUT data sets to use OUTPUT specifications.
JESDS=ALL	directs system data sets of output to use OUTPUT specifications.

"tt"	specifies the number of lines per page before an automatic eject is to occur. The default is to fill all but a half inch margin at the top and bottom of the page (e.g., at 8 lines per inch on 8-1/2 inch paper, 60 lines).
TRC=YES	specifies that the output contains Table Reference characters for producing multiple fonts.

Two parameters are required when the OUTPUT statement is used, FORMDEF and FORM. The *OS/390 MVS JCL User's Guide* (GC28-1758) contains a full description of all of the valid parameters of the OUTPUT JCL statement.

Since the OUTPUT statement is supported only for cut-sheet printing at the NIH Computer Center, the following parameters are not supported:

BURST	CKPTLIN	CKPTPAGE	CKPTSEC
CLASS	COMPACT	CONTROL	DATAACK
DEST	FLASH	GROUPID	INDEX
LINDEX	MODIFY	PAGEDEF	PIMSG
PRMODE	PRTY	THRESHLD	UCS
WRITER			

The above parameters have not been tested in the NIH Computer Center environment and may produce unpredictable results if used. Complete details on the capabilities of the cut-sheet printers, the forms, the FORMDEFs, the FCBs, and character sets available, and the interactions between the various printing options are given in the *NIH Computer Center User's Guide*. The SYSOUT DD statements for a single data set are described in 2.5.2.

Examples

To cause an entire job (that is, all job data sets) to be printed double sided on form 900P on the cut-sheet laser printer:

```
//OUTALL OUTPUT FORMDEF=PD00,FORMS=900P,
//          JESDS=ALL,DEFAULT=YES
```

To cause all the output data sets to be printed single sided on form 999P on the cut-sheet laser printer, but have all system data sets from the job sent to the laser standard printer:

```
//OUTALL OUTPUT FORMDEF=PS99,FORMS=999P,DEFAULT=YES
```

To cause 25 copies of a single SYSOUT data set to be printed double sided on 900L paper, using character set LB12 at 6 lines per inch:

```

//DATA1 OUTPUT FORMDEF=LD00,FORMS=900L,CHARS=LB12,
//  FCB=806,COPIES=25
:
:
// EXEC ....
:
:
//SYSPRINT DD SYSOUT=A,OUTPUT=*.DATA1

```

To cause the output of a job to be printed on NIH letterhead forms:

```

//OUT1 OUTPUT FORMDEF=LH1,PAGEDEF=LH1,FORMS=900P
:
:
//SYSUT2 DD SYSOUT=A,
//  OUTPUT=( *.OUT1)

```

2.11.5 /*NOPURGE

The /*NOPURGE statement causes output of a job directed to the OUTPUT HOLD queue to be printed rather than purged when its retention period in the HOLD queue has expired. For example, it can be used to preserve output which may not be accessed before a weekday holiday.

```

/*NOPURGE

```

If a /*NOPURGE statement is included in a job which is not directed to the OUTPUT HOLD queue, it will be ignored.

2.11.6 Combining Printing Statements

The interaction between the three types of output statements is controlled by the following hierarchy:

- /*DDOUT takes precedence over SYSOUT DD.
- SYSOUT DD takes precedence over /*JOBOUT.
- /*JOBOUT overrides the standard defaults for the job's output.

Since the SYSOUT statement offers only a limited number of options, use of /*DDOUT to specify all options for a data set is recommended and will reduce the complexity of overriding.

Example

```
//JOB LIB DD DSN=aaaaaaa.LIBRARY,DISP=SHR
/*JOBOUT COPIES=3
/*DDOUT XXX FORMS=Fnnn,DEST=(R94,CENTRAL)
/*DDOUT ZZZ COPIES=1
//S1 EXEC PGM=MAIN
.
.
//OUT1 DD SYSOUT=(A, ,XXX)
.
.
//ERRS DD SYSOUT=(A, ,ZZZ)
.
.
//OUT2 DD SYSOUT=A
.
.
//OUT3 DD SYSOUT=(A, ,XXX)
```

The /*JOBOUT statement causes three copies of the job's system output to be printed at the central facility using laser standard paper and the SN12 character set.

The /*DDOUT statements affect the OUT1, OUT3, and ERRS SYSOUT data sets.

The SYSOUT output will be handled as follows:

- | | |
|------|---|
| OUT1 | Three copies will be printed with the SN10 character set on user-supplied forms at the central facility and three copies will be printed on user-supplied forms at Remote 94. (If Remote 94 is not supplied with the specified forms, the listing will be held for printing until the remote's operator releases it.) |
| ERRS | One copy will be printed using the default laser standard characteristics at the central facility. |
| OUT2 | Three copies will be printed with the SN12 character set on laser standard paper at the central facility. |
| OUT3 | is the same as OUT1. |

/*JOBOUT and /*DDOUT statements may both be specified for a job. The /*JOBOUT statement is used to establish new printing defaults for the job, and /*DDOUT statements are used to specify specific requirements for those SYSOUT DD statements that should not use these defaults. The following would cause three copies of all job output, except the SYSUDUMP data sets, to print on 999P paper (with the SN10 character set and at a density of 6 lines-per-inch). Only a single copy of any SYSUDUMP output will be printed, using the default, laser standard printing characteristics.

```
/*JOBOUT   FORMS=999P,COPIES=3
/*DDOUT    ABC FORMS=STD,COPIES=1
:
//PRT1     DD SYSOUT=A
:
//GO.SYSUDUMP DD SYSOUT=(A, ,ABC)
:
//STUFF5   DD  SYSOUT=A
:
//SYSUDUMP DD  SYSOUT=(A, ,ABC)
```

2.12 PRINTING CONTROL STATEMENTS

For an overview of the printing services offered at the NIH Computer Center, see the *NIH Computer Center User's Guide*.

2.12.1 /*TITLE

To print a title on the header and trailer pages of a batch job listing, include the following statement:

```
/*TITLE=title
```

Where "title" is up to 20 characters beginning with the first non-blank character after column 8.

If more than 20 characters are given in the title, only the first 20 are used. The equal sign is optional; a blank may be substituted. If more than one /*TITLE statement is present, the last encountered will be used.

Examples

```
/*TITLE MASTER LISTING

/*TITLE=April 8, 2001

/*TITLE  Building 12A - 1011
```

2.12.2 WYLBUR Printing Control Statements

The following printing control statements apply only to WYLBUR and are documented in the *WYLBUR Batch Processing* manual:

```
/*DATA
/*IBM
/*INSERT
/*MERGE
/*NUMBERED
/*OVERLAY
```

```
/*TSO
/*UNNUMBERED
```

2.13 COMMENT STATEMENT

A comment statement may be inserted anywhere after the JOB statement, including between the sections of a continued statement.

```
// * comment
```

The comment statement consists of the characters `/*` in columns 1-3. The comment can be typed anywhere in columns 4-80.

Examples

```
/*DATA MUST FOLLOW FORMAT 1

/* *** CALL TO INDEX ROUTINE ***
```

2.14 AVOIDING JOB TURNAROUND DELAYS

Users who have become accustomed to the usual levels of turnaround at the NIH Computer Center are concerned when turnaround increases and wonder about the cause. Delays can be caused by equipment malfunctions or system problems. Such delays are, by their nature, unpredictable, but there are other causes for delay which can be understood and perhaps avoided.

A batch job goes through several stages before it actually begins to execute the programs in the job. In the first stage, "AWAITING EXECUTION," the job is simply waiting for the operating system to select the job for execution. Very few system resources are used during this stage. Once the operating system has selected the job, it begins the "EXECUTING" stage. At this point, cross-system enqueue resolves data set contentions and system resources start being consumed.

Delays AWAITING EXECUTION

- Tape handling delays the start of execution until the tape has been fetched from the library or the operator has ascertained that the special tape has been received. (This is the reason Class B jobs take longer than Class E even though they have stricter time limits.)
- RECOVER jobs that restore data sets from backup tapes of private data sets require exclusive use of an intermediate system disk so only one can execute at a time.
- `/*BEFORE`, `/*AFTER`, and `/*CONTROL` statements cause jobs to wait longer than normal to avoid conflicts.
- `/*DISCOUNT` and `/*ROUTE XEQ LATE` prevent jobs from executing until the designated time period.

-
- Multiple jobs with the same jobname cannot execute simultaneously on the same processor. Giving each job a unique jobname will avoid this problem.

Long ELAPSED TIME for Execution

- Data set contention occurs when multiple jobs or tasks require conflicting access to one or more data sets. It is indicated by MIM1038 and MIM1039 messages in the JES2 JOB LOG. Until all data set contentions for every step of a job are resolved, the job will never actually begin execution. An "EXECUTING" job can, in turn, cause "EXECUTING" jobs after it to wait for the same or additional data sets. This process can create long "waiting lines" of jobs, with many jobs apparently "EXECUTING," but actually just sitting suspended in the system waiting for data sets. These "waiting lines" prevent the system from selecting other jobs from the "AWAITING EXECUTION" queue. The batch jobs that are suspended waiting for access to the data set will be tying up system resources, yet doing no productive work. Any job waiting for access will be cancelled after 30 minutes in execution suspension. Ways to avoid data set contention include:
 - Using DISP=SHR whenever a data set is to be read and not modified.
 - Using unique data set names for work (scratch) data sets and for data sets on different volumes.
 - Using /*BEFORE, /*AFTER, and /*CNTL statements to reduce contention problems.
 - Using DSSUBMIT to execute a group of jobs in a pre-determined order.
- BDAM (direct) file processing, when OPTCD=E (extended search) was used to create the file and the LIMCT DCB subparameter is not used to describe the file, can cause a single random search for one record to search the entire file before the "key-not-found" condition is signaled back to the processing program.

2.15 INTERACTIVE FACILITIES FOR BATCH JOB SUBMISSION

Batch jobs can be submitted interactively through the Web Submit facility and through WYLBUR's ENTER SUBMIT command (to request the scheduled automatic submission of recurring batch jobs).

2.15.1 Batch Job Submission through the Web

Users can submit their batch jobs through the World Wide Web using the Web Submit facility of SILK (Secure Internet LinKed) technologies. Open your Web browser software to:

<http://silk.nih.gov/submit>

Enter a data set name to be submitted as a batch job and specify options—including remote print routing, impact printing, DISCOUNT, HOLD, NOTIFY, as well as BEFORE, AFTER or CNTL.

Supplied by the System

- If the job name of the submitted job begins with the initials specified (in the account/initials field of the Web browser security window), then the appropriate keyword will be inserted.
- If the RACFid in the job statement (USER=racfid) is the same as the specified initials, then the proper password will be supplied.

Supplied by the User

- If the job name begins with different initials, supply the keyword in the Web "Submit" form, or have a /*KEYWORD statement in the data set being submitted.
- If the RACFid in the job statement (USER=racfid) is not the same as the specified initials, supply the password on the Web form or have a /*PASSWORD statement in the data set being submitted

The data set being submitted via SILK Web's submit button **must be FB with LRECL=80**. Please note: the data set can **not** be WYLBUR EDIT format. If the data set is RACF protected, the initials specified in the browser security window must have authority at least to "read" that data set. After you click on the submit button, the data set will be submitted and the job number will be displayed.

If the job was rejected for some reason, that information will also be displayed. To find why the job was rejected, look at the printed output or fetch the job through WYLBUR or TSO.

For further information on SILK Web technologies, refer to the *NIH Computer Center User's Guide*, or visit:

<http://silk.nih.gov>

2.15.2 Automatic Scheduled Batch Job Submission through WYLBUR

WYLBUR's ENTER SUBMIT command can be used to request that a batch job be submitted automatically at a specific time every day, every weekday, one day per week, on the first weekday of each month, the last weekday of each month, a specific day (e.g., the 22nd) of each month, or on one given date. ENTER SUBMIT also includes an option to suspend the running of jobs on holidays. The user simply supplies the name of a cataloged data set containing the JCL, including a /*KEYWORD statement (and /*PASSWORD if necessary), that is to be submitted. Information as to whether the job is to be submitted every day, every weekday, one day per week, or on only one specific date will be requested by the ENTER SUBMIT command procedure.

With ENTER SUBMIT, a program can be run at a specific time and date, giving greater control than /*DISCOUNT and /*LATE. For full details on ENTER SUBMIT, see the *NIH Computer Center User's Guide*.

2.16 UNDERSTANDING JOB OUTPUT

This section gives an elementary explanation designed to help beginners find their way through the output of a job. It is followed by a sample job output. There are six sections to the output of any job:

Header sheet (not illustrated) For jobs printed at the central facility, this has CIT's address. The number of the user's outbox is shown in large numerals. The job number, the date the job was submitted, and the programmer's name are listed to the left of the large numerals. The jobname, printer number, form number, and submission time are listed to the right. Below the CIT logo, the information is repeated. If the job was submitted through WYLBUR, WYLxxx is included. The printer number and the box number are shown.

When the job is printed at a remote printer the following information appears: job number, jobname, date, submission time, programmer's initials, programmer's name, printer number, box number and CIT's address.

JES2 job log To the left of each line on this page is a 24 hour clock time (of the pattern: hour.minute.second). The \$HASP373 message shows the time the job started execution, the job number, the job name, the initiator used, the job class, subsystem it ran on, and the date it was run.

The final line is the \$HASP395 message which shows the time the job ended. Between these there may be other system messages.

If there were an error in the job, the job log would include error messages indicating an ABEND, JCL error, or hardware or software failure.

Expanded JCL This page begins with the JOB statement and job number. It lists both the JCL statements submitted by the user (preceded by //) and the JCL statements from invoked procedures (preceded by XX) as well as the messages which show the symbolic substitutions performed. Any statements in procedures that are overridden by user-supplied statements are preceded by X/. The JCL statements are numbered to the left so they can be referred to by later messages.

Operating system information This section contains the messages generated by the system for each job step. These messages give such information as unit allocations and deallocations, data set dispositions, volume serial numbers, and step completion codes. Any JCL errors are also noted in the step where the error occurred. The number to the left of each message is the number of the JCL statement it refers to in the expanded JCL listing.

A step accounting block (surrounded by asterisks) is included in this section for each job step. The sample job has two EXEC statements, therefore it has two job steps.

The first line of the step accounting block gives the times the step started and ended and the step completion code. An S or U indicates a system or user ABEND. C indicates a condition code and is followed by a number from 0 to 16. This number is determined by the program. A zero usually indicates there were no errors; the higher numbers usually indicate errors. (In the sample, "C 0" indicates the step completed successfully).

The second line gives the total step CPU time. The next two lines list chargeable resources used by the step. The final line of the block gives the STEPNAME and STEP NUMBER as well as the cost for the step.

The step accounting block also includes input/output operations for each step. The UNIT field corresponds to the physical address on which the disk or tape volume resides. The D/T field indicates whether the DDNAME refers to disk or tape media. DDNAME corresponds to the name specified in the JCL DD statement. The I/O field reports the number of input/output operations (also known as EXCP's; one for each block of data read or written) for a particular DDNAME. DDNAMES that are allocated but have no I/O operations are not reported.

After the last step accounting block come the job deallocation messages issued when the system cleans up data sets which are still allocated at the end of the job. They are followed by the job accounting block, which has asterisks around it. The third line of this block identifies the computer on which the job was run.

The job accounting block also includes the JOB COMPLETION CODE. (In the sample, "C 0" indicates the job finished successfully. Since this is a multi-step job, this code indicates the completion code for the final step that was executed.) Job identification information, total CPU time and lines printed for all steps are also listed. The next-to-last line gives the total cost for running the job. Printing costs are reported on a separate page just before the trailer sheet.

After the last step accounting block come the job deallocation messages issued when the system cleans up data sets that are still allocated at job end. They are followed by the job accounting block, which has asterisks around it. The third line of this block identifies the computer on which the job was run.

Program output	The contents of this section vary depending on the nature of the job. They might include: compiler output, Binder output, reports generated by the program, messages indicating whether the job ran successfully or not, and messages from utility programs. Since the sample is a simple copy, it does not include the compiler output and Binder output that would have been present if a language such as FORTRAN had been used.
Printing costs	This page reports the printing costs for the job.
Trailer sheet	This is similar to the cover sheet; the end time for the job indicates when the job was printed.

Figure 2-5. Sample Job Output

```

JES2 JOB LOG ***** NODE NIHC U

11.20.10 JOB03366 ICH70001I III LAST ACCESS AT 10:54:04 ON THURSDAY, NOVEMBER 8, 2001
11.20.10 JOB03366 $HASP373 III6273W STARTED - INIT 1 - CLASS B - SYS CPU6 - 08 Nov 2001
11.20.11 JOB03366 IEF285I AAAAIII.NEW.COPY.DATA KEPT
11.20.11 JOB03366 IEF285I VOL SER NOS= NIHLBD.
11.20.11 JOB03366 IEF285I III.III6273W.JOB03366.D0000101.? SYSOUT
11.20.11 JOB03366 IEF285I III.III6273W.JOB03366.D0000102.? SYSOUT
11.20.11 JOB03366 IEF285I SYS01312.T112011.RA000.III6273W.R0183829 KEPT
11.20.11 JOB03366 IEF285I VOL SER NOS= DSA137.
11.20.12 JOB03366 IEF285I III.III6273W.JOB03366.D0000103.? SYSOUT
11.20.12 JOB03366 $HASP395 III6273W ENDED

1 //III6273W JOB (AAAA,999,B,,,),'CITUSER',USER=III JOB03366
/*PASSWORD **JES2**
/* JOB SUBMITTED BY III USING WYLBUR
/*KEYWORD= **JES2**
/*ROUTE PRINT HOLD **JES2**
/*NOTIFY III **JES2**
2 //SCRSTEP EXEC DSSCR,NAME=AAAAIII.NEW.COPY.DATA 2.
XX*
XX* SCRATCHES A DATA SET STORED ON DISK
XX*
XX* SYS - 04/20/87
XX*
3 XX PROC NAME=, (REQ) DATA SET TO BE SCRATCHED
XX STORAGE=FILE, DISK UNIT
XX DISK= DISK VOLUME SERIAL
XX*
4 XXDSSCR EXEC PGM=DSSCR,REGION=512K,PARM='VOL="&DISK",UNIT="&STORAGE"'
XX*
IEFC653I SUBSTITUTION JCL - PGM=DSSCR,REGION=512K,PARM='VOL="",UNIT="FILE"'
5 XXDSNAME DD DSNAME=&NAME,
XX VOL=REF=NIH.EMPTY,DISP=OLD
IEFC653I SUBSTITUTION JCL - DSNAME=AAAAIII.NEW.COPY.DATA,VOL=REF=NIH.EMPTY,DISP=OLD
6 XXSYSPRINT DD SYSOUT=*
7 XXSYSUDUMP DD SYSOUT=*
8 //STEP1 EXEC COPY 3.
XX*
XX* COPIES A SEQUENTIAL DATA SET
XX*
XX* CFB - 05/12/97
XX*
9 XXCOPY EXEC PGM=IEBGENER
XX*
10 XXSYSPRINT DD SYSOUT=*
11 //COPY.SYSUT1 DD DSN=AAAAIII.WEEKDAY.LIST,UNIT=FILE, 4.
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440) 5.
X/SYSUT1 DD DISP=SHR
12 //COPY.SYSUT2 DD DSN=AAAAIII.NEW.COPY.DATA,UNIT=FILE, 6.
// SPACE=(TRK,(5,1),RLSE),DISP=(NEW,CATLG) 7.
X/SYSUT2 DD DISP=(NEW,KEEP,DELETE)
13 XXSYSIN DD DUMMY
STMT NO. MESSAGE
2 IEFC001I PROCEDURE DSSCR WAS EXPANDED USING SYSTEM LIBRARY NIH.VPROCLIB
8 IEFC001I PROCEDURE COPY WAS EXPANDED USING SYSTEM LIBRARY NIH.VPROCLIB
ICH70001I III LAST ACCESS AT 10:54:04 ON THURSDAY, NOVEMBER 8, 2001
IEF236I ALLOC. FOR III6273W DSSCR SCRSTEP
IEF237I 0320 ALLOCATED TO DSNAME
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF237I JES2 ALLOCATED TO SYSUDUMP
IEF237I 0E02 ALLOCATED TO SYS00001
IEF142I III6273W DSSCR SCRSTEP - STEP WAS EXECUTED - COND CODE 0000
IEF285I AAAAIII.NEW.COPY.DATA KEPT
IEF285I VOL SER NOS= NIHLBD.
IEF285I III.III6273W.JOB03366.D0000101.? SYSOUT
IEF285I III.III6273W.JOB03366.D0000102.? SYSOUT
IEF285I SYS01312.T112011.RA000.III6273W.R0183829 KEPT
IEF285I VOL SER NOS= DSA137.

```

```

*****
* STEP START TIME - 11:20:11                STEP END TIME - 11:20:12                STEP COMPLETION CODE - C 0 *
* TOTAL STEP CPU TIME - 0.02 SECONDS          VECTOR CPU TIME - 0.00 SECONDS          PERCENT VECTOR CPU TIME - 0.0 *
* TAPES - 0                                REGION (REQUESTED - 512K USED - 44K)          I/O S - 0 *
* INPUT RECORDS SPOOLED - 12                CARDS SPOOLED FOR OUTPUT - 0          LINES SPOOLED FOR OUTPUT - 47 *
* STEPNAME - DSSCR                        CHARGE (DOLLARS - $.03 MACHINE UNITS - .150)          STEP NUMBER - 1 *
*****
*                                     DISK AND TAPE I/Os ARE AS FOLLOWS: *
* UNIT D/T DDNAME I/Os | UNIT D/T DDNAME I/Os | UNIT D/T DDNAME I/Os *
-----
IEF373I STEP/DSSCR /START 2001312.1120
IEF374I STEP/DSSCR /STOP 2001312.1120 CPU 0MIN 00.02SEC SRB 0MIN 00.00SEC VIRT 40K SYS 308K EXT 4K SYS 9948K
IEF236I ALLOC. FOR III6273W COPY STEP1
IEF237I JES2 ALLOCATED TO SYSPRINT
IGD103I SMS ALLOCATED TO DDNAME SYSUT1
IGD101I SMS ALLOCATED TO DDNAME (SYSUT2 )
        DSN (AAAAIII.NEW.COPY.DATA )
        STORCLAS (STANDARD) MGMTCLAS (FILEM3) DATACLAS (PS)
        VOL SER NOS= DSA125
IEF237I DMY ALLOCATED TO SYSIN
IEF142I III6273W COPY STEP1 - STEP WAS EXECUTED - COND CODE 0000
IEF285I III.III6273W.JOB03366.D0000103.?          SYSOUT
IGD104I AAAAIII.WEEKDAY.LIST                      RETAINED, DDNAME=SYSUT1
IGD104I AAAAIII.NEW.COPY.DATA                      RETAINED, DDNAME=SYSUT2
*****
* STEP START TIME - 11:20:12                STEP END TIME - 11:20:12                STEP COMPLETION CODE - C 0 *
* TOTAL STEP CPU TIME - 0.02 SECONDS          VECTOR CPU TIME - 0.00 SECONDS          PERCENT VECTOR CPU TIME - 0.0 *
* TAPES - 0                                REGION (REQUESTED - 4096K USED - 248K)          I/O S - 3 *
* INPUT RECORDS SPOOLED - 0                CARDS SPOOLED FOR OUTPUT - 0          LINES SPOOLED FOR OUTPUT - 4 *
* STEPNAME - COPY                        CHARGE (DOLLARS - $.03 MACHINE UNITS - .146)          STEP NUMBER - 2 *
*****
*                                     DISK AND TAPE I/Os ARE AS FOLLOWS: *
* UNIT D/T DDNAME I/Os | UNIT D/T DDNAME I/Os | UNIT D/T DDNAME I/Os *
-----
* 0782 D SYSUT1 2 | 07AA D SYSUT2 1 |
*****
IEF373I STEP/COPY /START 2001312.1120
IEF374I STEP/COPY /STOP 2001312.1120 CPU 0MIN 00.02SEC SRB 0MIN 00.00SEC VIRT 244K SYS 288K EXT 4K SYS 9984K
*****
*
*                                     MVS/ESA
*
*                                     NATIONAL INSTITUTES OF HEALTH
*
*                                     CENTER FOR INFORMATION TECHNOLOGY
*
*                                     NIH COMPUTER CENTER
*
*****
*
* PROGRAMMER - CITUSER                        DATE - 11/08/01                        JOB NAME - III6273W *
*
* TOTAL JOB CPU TIME - 0.04 SECONDS          JOB VECTOR CPU TIME - 0.00 SECONDS *
*
* ACCOUNT NUMBER - AAAA                        COMPUTER -IBM 9672 - 6 *
*
* INITIAL TIME - 11:20:10                    FINAL TIME - 11:20:12                    JOB COMPLETION CODE - C 0 *
*
* INPUT RECORDS SPOOLED - 12                CARDS SPOOLED FOR OUTPUT - 0          LINES SPOOLED FOR OUTPUT - 51 *
*
* CHARGE (DOLLARS - $.25 MACHINE UNITS - .296) *
*
* APPROXIMATELY EQUIVALENT 3090J CPU TIME - 2 SECONDS *
*
* CHARGES FOR PRINTING DONE AT THE CENTRAL FACILITY ARE SHOWN ON THE LAST PRINTED OUTPUT SHEET *
*****
IEF375I JOB/III6273W/START 2001312.1120
IEF376I JOB/III6273W/STOP 2001312.1120 CPU 0MIN 00.04SEC SRB 0MIN 00.00SEC
DSSCR11 - SCRATCH SUCCESSFUL FOR AAAAIII.NEW.COPY.DATA ON DSA137
DSSCR12 - UNCATALOG SUCCESSFUL FOR AAAAIII.NEW.COPY.DATA
DATA SET UTILITY - GENERATE
IEB352I WARNING : OUTPUT RECFM/LRECL/BLKSIZE COPIED FROM INPUT
PAGE 0001
PROCESSING ENDED AT EOD

*** PRINTING CHARGES ***
PAGES PRINTING: 0
LINES PRINTED: 169
MACHINE UNIT: 5.406
CHARGE: $1.00

```

3 SOUTH – SERVICE AND UTILITY PROGRAMS

Most South System cataloged procedures will be available on Titan, so it is expected that most users' jobs will run on Titan with little or no change. For more information on the Titan transition, go to:

<http://silk.nih.gov/silk/titan>

This section describes some of the utility programs available at the NIH Computer Center. These include IBM utilities, utility programs from other installations, and CIT-written utilities. Cataloged procedures have been set up for some of these programs to make certain common requests easier to perform. The description of each cataloged procedure or program includes instructions on its use and some of the functions available. These descriptions are not meant to be replacements for the appropriate manuals when they are available.

Figure 3-1. Directory of Utilities

ADSERASE, 133	JCLSCAN 227
ADSMAP, 130	MEMADD, 176
ADSRECOV, 201	MEMALTER, 178
CATDS, 102	MEMRENAM, 177
COMPARE, 123	MERGE, 122
CONNECT:Direct, 238	PALTAPE, 141
COPY, 105	PASBILL, 228
DDSCOPY, 190	PDSALLOC, 174
DDSGET, 167	PDSCOPY, 182
DDSLIST, 189	PDSDEL, 175
DDSSAVE, 166	PDSGET, 164
DISKEXPT, 200	PDSLST, 186
DISKGET, 213	PDSMAP, 180
DISKMAP, 194	PDSPACK, 181
DISKSAVE, 199	PDSSAVE, 166
DSCOPY, 107	POSTSCRIP, 116
DSGET, 157	PRINT, 110
DSLST, 113	RCVFILE, 234
DSRENAME, 170	RESPOUT, 143
DSSAVE, 156	RMTDSIN, 126
DSSCM, 124	RMTDSOUT, 128
DSSCMI, 125	SENDFILE, 234
DSSCR, 171	SORT, 119
DSSCRV, 172	TAPECOPY, 142
DSSUBMIT, 226	TAPEMAP, 139

DSUNSCM, 124	TAPESCAN, 149
EDSUTIL, 135	UNCATDS, 103
DSUNSCMI, 125	USRPDSC, 184
EDSGET, 161	USRPDSL, 180
EDSIN, 217	VDSUTIL, 103
EDSLIST, 114	VOLSTAT, 144
EDSOUT, 219	WYLBUR, 231
EDSSAVE, 159	WYLBURL, 231
EDSSORT, 120	

3.1 UTILITY PUBLICATIONS

Manuals relating to the topics discussed in this section can be ordered from CIT. For information on ordering publications, see Section 1.

Refer to the manuals in the following CIT categories:

- Computer Center User's Guides
- Batch Processing and JCL
- IBM Utilities
- Operating System
- VSAM
- Tapes
- TSO
- WYLBUR

Changes that affect job control language, batch processing, and utility programs will be fully tested and pre-announced through *Interface*, the technical newsletter published by the NIH Computer Center. For a full description of all NIH Computer Center services and facilities, refer to the *NIH Computer Center User's Guide*.

3.2 UTILITY RESTRICTIONS

In order to prevent accidental modification to the system and to safeguard customer data, there are restrictions on the use of some utility programs.

Use of the following IBM system utilities is forbidden:

IEHATLAS	IEBIMAGE
DSF	IFHSTATR
IEHINITT	ICAPRTBL

The following IBM system utilities and service aids should not be used to access any of the public DASD volumes. There are NIH Computer Center utilities to provide each of these functions.

AMASPZAP	(use MEMALTER)
IEHMOVE	(use PDSSAVE, PDSGET, DDSSAVE, DDSGET)
IEHPROGM	(use CATDS, UNCATDS)
IEHLIST	(use PDSMAP)

All backup and recovery of direct access volumes must be performed through facilities provided by the NIH Computer Center. The IBM dump/restore utility, DF/DSS, may not be invoked directly. Use the NIH Computer Center utilities DISKSAVE and DISKGET.

These IBM data set utilities may be used to access data on public and private volumes:

IEBCOMPR	IEBGENER
IEBCOPY	IEBISAM
IEBDG	IEBPTPCH
IEBEDIT	IEBUPDTE

The IBM data set utilities do not officially support processing of ASCII tape data sets. The NIH Computer Center's PALTAPE procedure has special facilities for handling ASCII tapes.

While the NIH Computer Center has no restrictions on dynamically invoking the utilities, it will only support the interfaces published in the documentation provided by the NIH Computer Center. Problems that develop in application systems that invoke these routines must be isolated to a specific, manageable level before assistance in problem diagnosis and resolution can be provided.

Users should not develop software that depends on particular output formats from NIH Computer Center utilities. SYSOUT changes are not always announced in advance.

Please note: Formal action will be taken against anyone who attempts to circumvent protection software and threatens the account and data security of the system.

3.3 MANIPULATING CATALOGED DATA SETS

The system catalog is a data set of information maintained by the operating system that is used to keep track of the location of data sets. Entries in the catalog include the name of the data set, the volume on which the data set is stored, and the type of device on which the data set is stored (e.g., a 3390 disk or a magnetic tape).

Any program can reference a cataloged data set (i.e., a data set whose name is in the catalog) by giving only its name. The system looks at the catalog to find out where the data set is stored, so the user doesn't need to specify the volume on which the data set is stored. This can be particularly useful when sharing a data set with other people. If other users are

accessing the data set by using the catalog, then you can change the location of the data set without informing the other users of the new location.

There are restrictions on the names of data sets:

- At NIH, data sets must follow the convention of beginning with an account/initials pair. (The high-level index must be of the form aaaaiii.)
- IBM standard naming conventions must also be followed. That is, periods must be used to divide the data set name into groups of eight or fewer characters, and each group of eight or fewer characters must begin with an alphabetic or national (@, \$, #) character.
- Only one data set with a given name can be cataloged.

Most data sets are cataloged or uncataloged by means of disposition processing, WYLBUR commands, or TSO commands. The CATDS procedure can be used for all data sets. UNCATDS can be used for data sets stored on tapes.

The WYLBUR commands SHOW DSNAMES ON CAT and the TSO command LISTCAT can be used to obtain information about cataloged data sets.

The DSNNAME and DISP parameters are always required to access a cataloged data set. The LABEL parameter is required only if the volume the data set resides on does not have a standard label. DCB parameters are required only if the processing program or the label does not supply full DCB information. Do not supply volume serial numbers or device types when accessing data sets through the catalog.

3.3.1 CATDS

The procedure CATDS executes the IBM utility program IEHPROGM, to perform cataloging and uncataloging functions. Data sets on public disk volumes must be cataloged. CATDS can only be used for data sets on tape or dedicated disk volumes.

These functions can also be accomplished with the WYLBUR and TSO CATALOG and UNCATALOG commands. The step name for the CATDS procedure is SCRATCH. No DD statements for the data sets are required to execute this procedure. The control statement for the catalog function has the format:

```
Col.2
      CATLG      DSNNAME=aaaaiii.name,VOL=device type=serial
```

DSNNAME and VOL must be spelled exactly as shown. "device type" for this procedure must be a generic class designation, i.e.,

3480	18-track 38000 BPI tape cartridge
3400-6	9-track 6250 or 1600 BPI tape
3390	online 3390 disk

(Classes like FILE and TAPE are not permitted.)

Example 1

```
//stepname EXEC CATDS
//SYSIN DD *
CATLG DSN=aaaaaaa.name1,VOL=3480=serial1
CATLG DSN=aaaaaaa.name2,VOL=3400-6=(serial3,1,serial4,1)
CATLG DSN=aaaaaaa.name3,VOL=3400-6=(serial5,1)
CATLG DSN=aaaaaaa.name4,VOL=3400-6=(serial5,2)
```

The data set to be cataloged by the second CATLG statement extends over 2 volumes, so the serial numbers must be enclosed in parentheses and listed in order of creation. The third and fourth CATLG statements illustrate a catalog operation for two data sets residing on the same tape. In this case, the data set sequence number must be given after the serial number, i.e., VOL=device=(serial,seqno).

3.3.2 UNCATDS

The procedure, UNCATDS, is used to uncatalog data sets on tape. The control statement has the format:

```
Col.2
      UNCATLG DSN=aaaaaaa.name
```

DSNAME must be spelled as shown and must specify the fully qualified name of the data set. Volume information is not required when uncataloging a data set.

Example 2

```
//stepname EXEC UNCATDS
//SYSIN DD *
UNCATLG DSN=aaaaaaa.name1
UNCATLG DSN=aaaaaaa.name2
UNCATLG DSN=aaaaaaa.name3
UNCATLG DSN=aaaaaaa.name4
```

This procedure does not delete or scratch a data set; it merely removes the entry from the system catalog. If the data set is retrieved by the catalog and given a disposition of DELETE, it will also be uncataloged.

3.3.3 VDSUTIL

The procedure VDSUTIL scratches or renames multiple cataloged data sets in a single step. This provides an alternative to the DSSCR procedure which can scratch only a single data set at a time. The management class for multiple cataloged data sets can also be changed by using VDSUTIL.

The following example scratches three data sets using a 1-step batch job.

Example 3

```
//STEP1 EXEC VDSUTIL
//SYSIN DD *
DELETE aaaaiii.dataset1
DELETE aaaaiii.dataset2
DELETE aaaaiii.dataset3
SET MAXCC = 0
```

The last command in the VDSUTIL example (SET MAXCC = 0) causes VDSUTIL to generate a return code of zero even if it cannot scratch one or more of the data sets. This corresponds to the way DSSCR works: returning a zero return code regardless of whether the specified data sets are actually scratched. If the SET command were left out, then VDSUTIL would return a nonzero return code if one or more of the data sets could not be scratched.

The following example renames two data sets using a 1-step batch job.

Example 4

```
//STEP1 EXEC VDSUTIL
//SYSIN DD *
ALTER aaaaiii.dataset1 NEWNAME(aaaaiii.dataset1.rename)
ALTER aaaaiii.dataset2 NEWNAME(aaaaiii.dataset2.rename)
SET MAXCC = 0
```

The following example changes the management class for multiple data sets. The first data set is placed into the MSS management class; the second data set is placed into the TMP management class.

Example 5

```
//STEP1 EXEC VDSUTIL
//SYSIN DD *
ALTER aaaaiii.dataset1 MANAGEMENTCLASS(MSS)
ALTER aaaaiii.dataset2 MANAGEMENTCLASS(TMP)
SET MAXCC = 0
```

3.4 GENERAL DATA SET UTILITIES

The utilities described in this section can be used to handle both tape and disk data sets. For utilities that handle only disk data sets, see Section 3.6; for those that handle only tape data sets, see Section 3.5. The utilities used for printing data sets are summarized in a table in Section 3.4.3.

3.4.1 Summary of General Data Set Utilities

Figure 3-2. General Data Set Utilities

Procedure Name	Documentation*	Function	Input
COPY (IBM utility IEBGENER)	3.4.2.1 <i>IBM DFSMS/MVS Utilities</i> , SC26-4926	Create a copy of a data set or a sequential data set from a member of a PDS; permits editing.	Sequential or partitioned data set
DSCOPY	3.4.2.2	Create a copy of a data set or a sequential data set on tape or disk.	Sequential or partitioned data set
PRINT (IBM utility IEBTPCH)	3.4.3.2 <i>IBM DFSMS/MVS Utilities</i> , SC26-4926	Print one tape or partitioned data set. Can print character or hexadecimal format.	Sequential or partitioned data set
COMPARE (IBM utility IEBCOMPR)	3.4.5	Compare two data sets.	Two sequential data sets or PDSs
SORT (IBM program SORT)	3.4.4.1 <i>IBM DFSORT Application Programming Guide</i> , SC33-4035	Sort using 3-disk work areas.	Sequential data sets
MERGE (IBM program SORT)	3.4.4.3 <i>IBM DFSORT Application Programming Guide</i> , SC33-4035	Merge presorted data sets.	Up to 16 sorted sequential data sets
ADSERASE	3.4.9	Erase data.	Entire tapes or disk data sets

Note: *Section numbers refer to sections in *OS/390 Batch and Utilities*.

3.4.2 Copying Data Sets

This section describes the COPY and DSCOPY procedures.

3.4.2.1 COPY

The IBM utility IEBGENER (see Section 3.2) is used in the COPY procedure to create a copy of a sequential data set or to create a sequential data set from a member of a partitioned data set. However, COPY cannot be used to copy a member of a PDS that is a load module; for that, the USRPDSC procedure described in Section 3.6.5.10 is used. To create a sequential data set from a load module member, the IEBCOPY procedure must be used.

COPY also cannot be used with an input tape that contains ASCII variable length records. For help in copying such data, contact the CIT Technical Assistance and Support Center (TASC).

The input and output data sets may have fixed, variable, or undefined length records, and they may have different blocking factors. Whenever LRECL, RECFM, or BLKSIZE is coded for the output data set, all three must be given.

SYSUT1 is the ddname of the input data set; SYSUT2 is the output data set. If DCB information is not specified for the output data set, the DCB for the input data set is used and a warning message is issued. The defaults are DISP=SHR for SYSUT1, and DISP=(NEW,KEEP,DELETE) for SYSUT2.

In the following example the input and output data sets are cataloged data sets on public disks. Note that the blocking factor is changed.

Example 6

```
//stepname EXEC COPY
//COPY.SYSUT1 DD DSN=aaaaaaa.dsname1,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800)
//COPY.SYSUT2 DD DSN=aaaaaaa.dsname2,
// SPACE=(TRK,(5,1),RLSE),UNIT=FILE,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440),
// DISP=(NEW,CATLG)
```

The following example illustrates a copy of one member of a cataloged partitioned data set. The member will be copied as a sequential data set onto a pre-assigned tape. Only one member at a time can be copied with this method.

Example 7

```
//stepname EXEC COPY
//COPY.SYSUT1 DD DSN=aaaaaaa.dsname1(member),
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
//COPY.SYSUT2 DD UNIT=TAPE,VOL=(PRIVATE,SER=tapeser),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440),
// DSN=aaaaaaa.dsname2
```

In the following example the input data set, which is written in ASCII code and has a non-standard data set name, is on a 9-track, 1600 BPI, ANSI-labeled tape which is handed in at Output Distribution Services. The output data set, written in EBCDIC code, is on a standard-labeled scratch tape. The default dispositions are used. Note that the blocking factor is changed. The job must be run Class C to keep the scratch tape after the job ends.

Example 8

```
//stepname      EXEC      COPY
//COPY.SYSUT1 DD UNIT=9TRACKHI,VOL=(PRIVATE,SER=tapeser),
//  DSN='nonstandard.name',LABEL=(1,AL),
//  DCB=(OPTCD=Q,RECFM=FB,LRECL=80,BLKSIZE=800)
//COPY.SYSUT2 DD UNIT=TAPE,VOL=PRIVATE,DSN=aaaaiii.name,
//  DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440)
```

For further information on the functions of the utility, IEBGENER, see the *DFSMS/MVS Utilities* manual, SC26-4926.

3.4.2.2 DSCOPY

DSCOPY is a procedure that executes a NIH Computer Center-written program to copy physical sequential data sets. It cannot be used to change the output values for RECFM, LRECL, or BLKSIZE or to copy multi-volume data sets. The ddnames IN and OUT describe the input and output data sets.

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is specified.
------	---

Optional Symbolic Parameters

NEWNAME	Names the output data set.
INDISK	Supplies the input volume serial number. Required only for a non-cataloged data set.
INTAPE	Replaces INDISK if the input volume is a tape.
OUTTAPE	Replaces OUTDISK if the output volume is a tape.
INSTOR	Provides a unit name for input (such as 9TRACKHI, MSS, or TAPE); the default is FILE.
OUTSTOR	Provides a unit name for output (such as 9TRACKHI, MSS, or TAPE); the default is FILE.
UNITS	Gives the units for the SPACE parameter; the default is TRK (tracks).
SIZE	Gives the primary space allocation for a disk output data set; the default is 10.
INCR	Gives the number of units in each secondary allocation; the default is 5.

UNUSED	Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
INDEN	Specifies a code (3 or 4) to indicate the density for OUTDEN a tape.
INLBL	Defines the type of label on a tape. SL OUTLBL (standard-labeled) is the default. NL (unlabeled) may be specified, but DCB information must be supplied if the input tape is NL.
LRECL	Gives the logical record length for the input data set; it is required if INLBL=NL and should not be used otherwise.
BLKSIZE	Gives the blocksize for the input data set; it is required if INLBL=NL and should not be used otherwise.
RECFM	Gives the record format for the input data set; it is required if INLBL=NL and should not be used otherwise.
INSEQ	Specifies a sequence number for a tape data set; the OUTSEQ default is 1.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default. New data sets on public disks are automatically cataloged.
OUTDISK	Supplies the output volume serial number. Required only if the data set is to be written to a dedicated disk.

In the example below, a cataloged sequential data set will be copied to a new cataloged MSS data set.

Example 9

```
//stepname EXEC DSCOPY,NAME='aaaaiii.dsname1',
//      NEWNAME='aaaaiii.dsname2',OUTSTOR=MSS
```

The following example will copy a cataloged data set to the Managed Storage System. The data set is given a new name because the catalog will not permit duplicate entries.

Example 10

```
//stepname EXEC DSCOPY,NAME='aaaaiii.dsname1',  
//      OUTSTOR=MSS,UNITS=CYL,NEWNAME='aaaaiii.dsname2'
```

The following example will copy a data set from an unlabeled 9-track 1600 BPI tape from another computer center to a standard-labeled 3480 cartridge NIH Computer Center tape. The data set will be named. DCB characteristics are given for the input data set.

Example 11

```
//stepname EXEC DSCOPY,NAME='dsname1',INTAPE=tapeser,  
//      INLBL=NL,INSTOR=9TRACKHI,INDEN=3,  
//      NEWNAME='aaaaiii.dsname2',OUTTAPE=tapeser,  
//      OUTSTOR=TAPE,RECFM=FB,LRECL=80,BLKSIZE=11440
```

The following example will copy the first data set from an ASCII unlabeled 9-track 6250 BPI special tape to a FILE data set. The input DD statement of the DSCOPY procedure is being overridden to request conversion from ASCII to EBCDIC. The new data set will be cataloged.

Example 12

```
//stepname EXEC DSCOPY,NAME='aaaaiii.dsname',  
//      INSTOR=9TRACKHI,INDEN=4,INTAPE=tapeser,INLBL=NL,  
//      LRECL=80,RECFM=FB,BLKSIZE=11440,SIZE=25  
//IN DD DCB=OPTCD=Q
```

3.4.3 Printing Data Sets

The print utilities available at the NIH Computer Center are described in this section.

3.4.3.1 Summary of Print Utilities

Figure 3-3. Print Utilities

Procedure Name	Documentation*	Function	Input
PRINT (IBM utility IEBTPCH)	3.4.3.2 <i>DFSMS/MVS Utilities</i> , SC26-4926	Prints data set; can select and format material printed.	Sequential or partitioned data set
DSLIS	3.4.3.3	Prints any non-EDIT format data set from tape or disk.	Sequential data set
USRPDSL	3.6.5.7	Prints contents of PDS members or member names only, with space and directory blocks allocated and used. No editing.	Partitioned data set
EDSLIS	3.4.3.4	Prints data set from tape or disk.	EDIT format or other sequential data set
PDSLIS	3.6.6.1	Prints entire data set; no editing.	Partitioned data set
DDSLIS	3.6.6.2	Prints entire data set; no editing.	Direct data set
POSTSCR	3.4.3.5	Prints PostScript data set.	Sequential or partitioned data set
PALTAPE	<i>PALTAPE</i>	Not a production tool; to be used for problem tapes only.	Tapes with problems

Note: *Section numbers refer to Sections in *OS/390 Batch and Utilities*.

3.4.3.2 PRINT

The PRINT utility utilizes the IBM utility IEBTPCH. Most users find it more convenient to use one of the other utilities in this section (e.g., DSLIS, PDSLIS, or EDSLIS) than to use PRINT.

Under the standard format, each new logical record begins on a new line. The end of a logical record is marked by an asterisk and the end of a block by two asterisks. Each line contains up to 96 characters, divided into 12 groups of 8 characters each, separated by two blanks. Therefore, if the input to the basic print operation is:

THIS IS WHAT THE DEFAULT OUTPUT LOOKS LIKE

the output will be:

```
THIS IS WHAT THE DEFAULT OUTPUT LOOKS LI KE*
```

Unprintable characters appear as blanks. The data can be printed in character or hexadecimal format or converted from packed decimal.

The default DCB attributes of the SYSUT2 output data set are:
RECFM=FBA,LRECL=121,BLKSIZE=121

Note: the sum of the field lengths in any IEBTPCH record statement may not exceed the output logical record length minus 1. Also, the maximum line size is 144.

The DD statement SYSUT1 defines the input data set. The procedure assumes sequential input and no editing on the output data set. The stepname is PRINT.

For more information on the IEBTPCH utility control statements, see the *IBM DFSMS/MVS Utilities* manual, SC26-4926.

Example 13

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD UNIT=9TRACK,DISP=(OLD,KEEP),
//      DSN=aaaaaaa.dsname,VOL=(PRIVATE,SER=tapeser),
//      LABEL=(1,NL),
//      DCB=(RECFM=FB,LRECL=80,BLKSIZE=11440,DEN=4)
```

In this example the first data set on a 9-track unlabeled 6250 BPI tape will be dumped to the printer. Since there is no label, DCB information must be supplied. If a standard-labeled tape is used, the DCB information is not needed on the DD statement. The data will be printed in standard format.

Frequently users want to print a tape, editing the data to eliminate the blanks between each group of eight characters. The following example illustrates how to do this and start printing after the 250th record. In this example the FIELD parameter takes advantage of all the defaults for starting input and output locations. It will read the first 80 bytes of the input record and start writing in the first position on the print line.

Example 14

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD DSN=aaaaaaa.dsname,UNIT=TAPE,
//      DISP=(OLD,KEEP),VOL=(PRIVATE,SER=tapeser)
//PRINT.SYSIN DD *,DSN=
//      PRINT STRTAFT=250,MAXFLDS=1
//      RECORD FIELD=(80)
```

No editing is allowed when printing variable spanned (VS) or variable blocked spanned (VBS) records. These data sets must be printed with the 8 character, 2 blank format.

If the LRECL of the data set is greater than 144, editing can still be done if the user chooses as the LRECL a valid line size (no more than 144) which can be evenly divided into the blocksize. In the case of a data set with an LRECL of 260 and a blocksize of 5200, each logical record can be broken into 2 print lines of 130 characters each. To accomplish this, DCB parameters must be specified for the input (SYSUT1) data set and IEBPTPCH control statements must be provided. DCB parameters must also be provided for the output (SYSUT2) data set since they exceed the default (121 output record characters per line). Notice that the LRECL is increased by 1 to allow for the carriage control. The input data set is cataloged.

Example 15

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD DSN=aaaaiii.dsname,DISP=SHR,
//          DCB=(RECFM=FB,LRECL=130,BLKSIZE=5200)
//PRINT.SYSUT2 DD DCB=(RECFM=FBA,LRECL=131,BLKSIZE=131)
//PRINT.SYSIN DD *,DSN=
PRINT MAXFLDS=1
RECORD FIELD=(130)
```

This method may be used for data sets with either fixed or fixed blocked records. If the data set has fixed-length unblocked records, RECFM=FB must be specified on the SYSUT1 DD statement.

The following JCL will cause a tape data set to be printed in hexadecimal:

Example 16

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD UNIT=TAPE,DISP=(OLD,KEEP),
//          VOL=(PRIVATE,SER=tapeser),DSN=aaaaiii.dsname
//PRINT.SYSIN DD *,DSN=
PRINT TOTCONV=XE
```

The following example illustrates how to print members of a cataloged PDS:

Example 17

```
//stepname EXEC PRINT
//PRINT.SYSUT1 DD DISP=(OLD,KEEP),
//          DSN=aaaaiii.dsname
//PRINT.SYSIN DD *,DSN=
PRINT TYPORG=PO,MAXNAME=3,MAXFLDS=2
MEMBER NAME=DATA1
RECORD FIELD=(100,1,,1)
MEMBER NAME=DATA2
MEMBER NAME=SORCE
```

RECORD FIELD=(80,10,,20)

Three members of a partitioned data set are to be printed. DCB information will be retrieved from the volume table of contents on the disk. (The logical record length happens to be 100). The control statements identify the three members to be printed and the editing to be performed. The first member, DATA1, will be printed with one logical record, 100 bytes per line beginning with position 1 of the input record, starting in column 1 of the print line. The second member, DATA2, will be printed with no format control. The third member, SORCE, will have only 80 bytes of each record printed; the rest of the logical record is to be ignored. Processing begins with the tenth byte of the input record; the printed output begins in column 20 of the print line.

3.4.3.3 DSLIST

DSLIST invokes the EDSUTIL program to list any sequential data set. If the record length is greater than 132, the record will be continued on following lines until it has all been printed.

Required Symbolic Parameters

NAME Names the data set to be listed.

Optional Symbolic Parameters

DISK	Specifies the volume serial number of the disk on which the data set is stored. It is not required if the data set is cataloged.
TAPE	Replaces DISK if the input volume is a tape.
STORAGE	Gives the unit name for the input device; the default is FILE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for a tape data set; the default is 1.
LBL	Defines the type of label on the input tape. SL (standard-labeled) is the default.
LRECL	Gives the logical record length for the input data set; it is required if LBL=NL and should not be used otherwise.
BLKSIZE	Gives the blocksize for the input data set; it is required if LBL=NL and should not be used otherwise.
RECFM	Gives the record format for the input data set; it is required if LBL=NL and should not be used otherwise.

DDOUT	Supplies printing options (e.g., forms, copies), which will affect only the listing. Gives a code for the SYSOUT=A DD statement to reference a /*DDOUT statement which details the options.
OPTIONS	The options, which may be specified for the output listing described in the <i>Edit Format Utility Package</i> manual. For example, two of the options are CC, to use the first character of each record for ANSI carriage control and INDENT=10 to insert 10 blanks at the beginning of each line.
UTOILOPT	Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the <i>Edit Format Utility Package</i> manual.

To list a cataloged sequential data set from a disk volume, use the following job control language:

Example 18

```
//stepname EXEC DSLIST,NAME='aaaaiii.dsname'
```

The following example will list the second data set from a cartridge tape.

Example 19

```
//stepname EXEC DSLIST,NAME='aaaaiii.dsname',
//          TAPE=tapeser,STORAGE=TAPE,SEQ=2
```

3.4.3.4 EDSLIST

EDSLIST invokes the EDSUTIL program to list an EDIT format or sequential data set. If the logical record length is greater than 132, the record will be continued on following lines until it has all been printed.

Required Symbolic Parameters

NAME Names the data set to be listed.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk on which the data set is stored. It is not required if the data set is cataloged.

TAPE Replaces DISK if the input volume is a tape.

STORAGE Gives the unit name for the input device; the default is FILE.

DEN	Specifies a code (3 or 4) to indicate the density for 9-track tape.
SEQ	Gives the sequence number for a tape data set; the default is 1.
LBL	Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (non-labeled) may be specified.
LRECL	Gives the logical record length for the input data set; it is required if LBL=NL and should not be used otherwise.
BLKSIZE	Gives the blocksize for the input data set; it is required if LBL=NL and should not be used otherwise.
RECFM	Gives the record format for the input data set; it is required if LBL=NL and should not be used otherwise.
DDOUT	Supplies printing options (e.g., forms, copies), which will affect only the listing. Gives a code for the SYSOUT=A DD statement to reference a /*DDOUT statement which details the options.
OPTIONS	The options, which may be specified for the output listing described in the <i>Edit Format Utility Package</i> manual. For example, two of the options are CC, to use the first character of each record for ANSI carriage control and INDENT=10 to insert 10 blanks at the beginning of each line.
UTIOPT	Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the <i>Edit Format Utility Package</i> manual.

To list a cataloged EDIT format data set, use the following job control language:

Example 20

```
//stepname EXEC EDSLIST,NAME='aaaaiii.dsname'
```

The following example will list a cataloged EDIT format data set, which has ANSI carriage control characters in column 1. Ten blanks will be inserted at the beginning of each line.

Example 21

```
//stepname EXEC EDSLIST,NAME='aaaaiii.dsname',
//          OPTIONS='CC,INDENT=10'
```

3.4.3.5 POSTSCRIP for PostScript Printing

The POSTSCRIP procedure is used to print PostScript files on the cut-sheet printers. The PostScript file must be transferred to the OS/390 System in binary format and have a logical record length (LRECL) less than or equal to 256. If no printing options are specified, single-sided, 8 1/2- by 11-inch copy will be produced. See the *NIH Computer Center User's Guide* for further information.

Required Symbolic Parameters

NAME Supplies the name of the input data set.

Optional Symbolic Parameters

COPIES Specifies the number of copies of output desired.

FORMDEF=PDTP Requests double sided printing

FORMS Specifies the type of paper to be used for printing. The default is 900P. The other forms for the cut-sheet printer (900L, 999P, and 999L) are valid forms.

The following example will print 7 copies of a PostScript file on cut-sheet paper:

Example 22

```
// EXEC POSTSCRIP,NAME='aaaaiii.dsname',COPIES=7
```

The following example will print a PostScript file on both sides of cut-sheet three-hole punched paper:

Example 23

```
// EXEC POSTSCRIP,NAME='aaaaiii.dsname',FORMDEF=PDTP
```

Uploading PostScript Files for use with POSTSCRIP

PostScript files can be uploaded to the OS/390 System for printing using any method that permits binary transfer (e.g., via Kermit and FTP). Binary transfer is necessary because the mainframe PostScript driver requires that the input be in ASCII. The file must also be uploaded with an LRECL of 256 or less.

Fonts Available with POSTSCRIP

The PostScript format supports numerous fonts. The following are currently available with the IBM OS/390 PostScript printing facility:

/Courier	/Courier-Bold
/Courier-BoldOblique	/Courier-Oblique
/Helvetica	/Helvetica-Black

/Helvetica-BlackOblique	/Helvetica-Bold
/Helvetica-BoldOblique	/Helvetica-Light
/Helvetica-LightOblique	/Helvetica-Narrow
/Helvetica-Narrow-Bold	/Helvetica-Narrow-BoldOblique
/Helvetica-Narrow-Oblique	/Helvetica-Oblique
/NewCenturySchlbk-Bold	/NewCenturySchlbk-BoldItalic
/NewCenturySchlbk-Italic	/NewCenturySchlbk-Roman
/Palatino-Bold	/Palatino-BoldItalic
/Palatino-Italic	/Palatino-Roman
/Symbol	/Times-Bold
/Times-BoldItalic	/Times-Italic
/Times-Roman	

Notes on PostScript Printing

PostScript can be generated by many different word processing and desktop publishing software products on multiple platforms. Software on some platforms, however, produces PostScript files that do not conform 100% to PostScript standards. Consequently, it is not uncommon to find that PostScript files produced on one platform will not print on PostScript-compatible printers on another platform. Although the IBM PostScript printing facility works for the vast majority of uploaded PostScript files, it is not guaranteed to work for every PostScript file. For this reason, it is recommended that users run extensive tests before developing any large-scale application using PostScript printing.

3.4.4 Sorting and Merging Data Sets

The SORT and EDSSORT procedures provided by the NIH Computer Center use the IBM DFSORT program. The SORT procedure requests 200K region below the line and 500K above the line; EDSSORT requests 300K below the line and 500K above the line. Since the DFSORT program uses all the available region, use of the maximum permissible region may make a job unnecessarily expensive. The sort/merge CHECKPOINT/RESTART facility is not supported.

Before attempting to override parameters in any of these procedures or trying to exceed any of the limitations they impose, consult the *DFSORT Application Programming Guide*, SC33-4035. When calling the DFSORT utility, PGM=Sort must be used for any stand-alone sort not invoking a NIH Computer Center procedure. The following statement must be included for the sort messages if the DFSORT program is called from a language unless a NIH Computer Center procedure is being used:

Example 24

```
//stepname.SORTMSGs DD SYSOUT=A
```

The DFSORT program can process VSAM and sequential data sets. Blocked or unblocked, fixed or variable length records can be sorted.

The NIH Computer Center does not provide procedures for using tape sort work areas (intermediate storage) because using disk is cheaper and reduces the possibility of I/O errors. When necessary, very large jobs should be divided into smaller jobs so they can be accommodated within the scratch disk limit.

The SORTIN DD statement describes the input data sets; the SORTOUT DD statement describes the output data set.

Non-VSAM data sets on unlike devices may be concatenated as SORTIN. All data sets in the concatenation must have the same record length and record format. The block size may vary, but the data set with the largest block size must be specified on the first DD statement in the concatenation.

The SORT control statement begins after column 1 and has the format given below. The second example would be used if all fields to be sorted were of the same format.

```
Col                                         Col
1                                         72
  SORT  FIELDS=(col,len,format,order,col,len,format,order...),  X
          FILSZ=numrec
```

or

```
  SORT  FIELDS=(col,len,order,col,len,order...),FORMAT=format,  X
          FILSZ=numrec
```

where

- | | |
|----------|---|
| "col" | the beginning position of the control field, relative to the beginning of the record (for variable length records, the logical record includes the 4-byte count field at the beginning of each record). |
| "len" | the length of the control field. |
| "format" | the format of the data in the control field. <ul style="list-style-type: none">• CH=character• ZD=zoned decimal• PD=packed decimal• FI=fixed point• BI=binary• FL=floating point |

HINT: To sort on a numeric field that might contain blanks, specify the character (CH) format.

"order"	"A" for ascending order; "D" for descending order.
"numrec"	a count of the number of records to be sorted. The sort will terminate if this count does not agree with the number of records read. If the number is not known exactly, code FILSZ=Enumrec, "E" indicating estimate.

"col," "len," and "order" are given once for each field to be used to sort the data set. If the SORT statement is to be continued onto the next statement, it should be broken after a comma, and column 72 must contain a non-blank character. The continuation statement must start in column 16.

For more information on additional SORT program facilities or SORT control statements, see *DFSORT Application Programming Guide*, SC33-4035.

3.4.4.1 SORT

The SORT procedure will sort approximately 1,650,300 80-byte records, and uses 3 sort work areas on 3 scratch disks. The default space allocation for the SORTWRK areas is 6 cylinders primary and 7 cylinders secondary, i.e., SPACE=(CYL,(6,7),RLSE). The SORTWK01, SORTWK02, and SORTWK03 DD statements have a default value of UNIT=SYSDA. The step name is SORT.

Using recommended blocksizes (Section 2.5.4.1) is particularly important for data sets that will be sorted. SORT reads and writes data with EXCP processing rather than the standard access methods. While this technique permits great sorting efficiency for "reasonable" block sizes, it is quite inefficient when sorting data with small blocksizes (e.g., 400).

Example 25

```
//stepname EXEC SORT
//SORT.SORTIN DD DSN=&&dsname,DISP=(OLD,DELETE),
//      DCB=(RECFM=FB,LRECL=120,BLKSIZE=11400)
//SORT.SORTOUT DD UNIT=FILE,DISP=(,CATLG),DCB=*.SORTIN,
//      SPACE=(TRK,(20,1),RLSE),DSN=aaaaiii.dsname
//SORT.SYSIN DD *
      SORT FIELDS=(10,7,CH,A),FILSZ=1254
```

In this example SORTIN is a temporary data set passed from a previous step. It will be deleted at the end of this step. The records are fixed length, 120 characters long, 95 records to a block. The SORTOUT data set is to be placed in the FILE management class and cataloged with a primary allocation request of 20 tracks. DCB information is copied from the SORTIN DD statement. The records will be sorted in ascending order on a seven-byte field beginning in position 10. There are exactly 1254 records in the input data set. The sort will terminate if the number of records read in does not agree with this value. If this value were an approximation of the number of records in the data set, FILSZ=E1254 could be used.

3.4.4.2 EDSSORT

EDSSORT places a sorted copy of an EDIT format data set into another EDIT format data set. It can be used to sort alphabetical and numerical data. For small amounts of data, WYLBUR's SORT command will perform this same function.

Required Symbolic Parameters

NAME Names the input data set.

NEWNAME Names the sorted output data set.

Optional Symbolic Parameters

DISK Supplies the input volume serial number. Not required if the data set is cataloged.

NEWDISK Supplies the output volume serial number. Required only for dedicated disk files.

STORAGE Gives the unit name for the input data set; the default is FILE.

NEWSTOR Gives the unit name for the output data set; the default is FILE.

UNITS Gives the units for the SPACE parameter; the default is TRK (tracks).

SIZE Gives the primary allocation for the SPACE parameter; the default is 10.

INCR Gives the number of units in each secondary allocation; the default is 5.

LRECL Gives the length of the longest line for the input (and output) data sets. The default is 133; longer records will be truncated unless LRECL is specified.

BLKSIZE Specifies a block size: the default is 11438. The value given must be an even multiple of the LRECL.

UNUSED Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.

STATUS OLD indicates that the output data set already exists and should be written over; the default is NEW.

UTILOPT Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the *Edit Format Utility Package* manual.

STEPEND Indicates the final (end of step) disposition for the output data set; CATLG is the default.

Example 26

```
//stepname EXEC EDSSORT,NAME='aaaaiii.dsname',  
//          NEWNAME='aaaaiii.newdsname'  
//EDSSORT.SYSIN DD *  
SORT  FIELDS=(col,len,order,...),FORMAT=CH  
/*
```

where

"col" the starting column number of the field to be used to sort the lines in the data set.

"len" the length of the field.

"order" A for ascending order or D for descending order.

"col," "len," and "order" are given once for each field to be used to sort the data set. The SORT statement must begin after column 1. If it is to be continued onto the next statement, it should be broken after a comma and column 72 must contain a non-blank character. The continuation statement must start in column 16.

EDSSORT assumes that the new data set does not already exist. To write over an existing data set, use:

Example 27

```
//stepname EXEC EDSSORT,NAME='aaaaiii.dsname',  
//          NEWNAME='aaaaiii.newdsname',STATUS=OLD,  
//          STEPEND=CATLG  
//EDSSORT.SYSIN DD *  
SORT  FIELDS=(15,10,A),FORMAT=CH
```

The default amount of space allocated for the sorted file should be adequate for most applications, but it may be changed.

Example 28

```
//stepname EXEC EDSSORT,NAME='aaaaiii.dsname',  
//          NEWNAME='aaaaiii.newdsname',SIZE=primary,  
//          STEPEND=CATLG  
//EDSSORT.SYSIN DD *  
SORT  FIELDS=(25,3,A,5,2,D),FORMAT=CH
```

3.4.4.3 MERGE

The merge procedure will accept up to 16 sequential data sets as input. These data sets must already be ordered on the fields described in the MERGE control statement. The input ddnames must be SORTIN01 through SORTIN16, and they must be in ascending sequence. SORTOUT is the ddname for the output data set. The records may be blocked or unblocked, fixed or variable length, but for a given application the record formats must be the same. If unlike block sizes are to be merged, the largest block size must be first. The merge procedure runs in 200K. The stepname is MERGE.

The control statement format is the same as that of the SORT control statement, but the keyword is MERGE:

```
MERGE  FIELDS=(col, len, format, order, col, len, format, order...), X
        FILSZ=numrec
```

or

```
MERGE  FIELDS=(col, len, order, col, len, order...), FORMAT=format, X
        FILSZ=numrec
```

where

"col"	the beginning position of the control field, relative to the beginning of the record (for variable length records, the logical record includes the 4-byte count field at the beginning of each record).
"len"	the length of the control field.
"format"	<div>the format of the data in the control field.<ul style="list-style-type: none">• CH=character• ZD=zoned decimal• PD=packed decimal• FI=fixed point• BI=binary• FL=floating point</div>
"order"	"A" for ascending order; "D" for descending order.
"numrec"	a count of the number of records in all the input data sets which are to be merged. The merge will terminate if this count does not agree with the number of records read. If the number is not known exactly, code FILSZ=Enumrec, "E" indicating estimate.

Example 29

```
//stepname EXEC MERGE
//MERGE.SORTIN01 DD DSN=aaaaiii.dsname,UNIT=TAPE,
// VOL=(PRIVATE,SER=tapeser),DISP=SHR,
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=11438,DEN=4)
//MERGE.SORTIN02 DD DSN=&&dsname,DISP=(OLD,DELETE),
// DCB=*.SORTIN01
//MERGE.SORTOUT DD DSN=aaaaiii.dsname,UNIT=TAPE,
// DISP=(,KEEP),VOL=PRIVATE,DCB=*.SORTIN01
//MERGE.SYSIN DD *
MERGE FIELDS=(20,10,CH,D),FILSZ=E120000
END
```

In this example two data sets are to be merged. One is from a tape, and the other is a temporary disk data set that has been passed from a previous step and will be deleted at the end of this step. The merged output is placed on a scratch tape which will be kept at the end of this step if the job is run Class C. The input data sets will be merged in descending order on the ten-byte control field beginning in position 20 of each record. The total number of records from both input data sets is approximately 120,000.

3.4.5 COMPARE for Comparing Data Sets

The COMPARE procedure uses the IBM utility program, IEBCOMPR, to compare two sequential data sets at the logical level. Since the comparison operates at the logical record level, the two data sets can have different blocksizes and can reside on different device types. The logical record lengths must be identical.

Two data sets are considered equal if they contain the same number of records and the corresponding records are identical. On an unequal compare, record and block number and the unequal records are printed. Ten successive unequal comparisons will terminate the job step. The program can compare fixed-length, variable-length, or undefined-length records from blocked or unblocked data sets.

The procedure name is COMPARE, the step name is CMP. SYSUT1 and SYSUT2 are the ddnames of the input data sets. The defaults are DISP=SHR for SYSUT1 and SYSUT2.

Example 30

```
//stepname EXEC COMPARE
//CMP.SYSUT1 DD UNIT=TAPE,VOL=(PRIVATE,SER=tapeser),
// DSN=aaaaiii.dsname1
//CMP.SYSUT2 DD DSN=aaaaiii.dsname2
```

Two sequential data sets, one on tape and the other on a public volume, are being compared. DCB information is supplied by the labels and default dispositions are used.

3.4.6 Scrambling and Unscrambling Data Sets

Programs are available to permit the user to scramble and unscramble the contents of a sequential data set. Data sets with a variable blocked record format and entire partitioned data sets should not be scrambled (a member of a PDS may be scrambled). These programs scramble the complete physical block and thereby destroy control information in partitioned and VB data sets. The scrambling process is controlled by a code phrase that is chosen by the user. A scrambled data set will occupy the same amount of space as it did before it was scrambled.

Cataloged procedures have been provided which will scramble a specified data set by creating a new data set or by updating the data set in place. The scrambling procedures use a code phrase chosen by the user to create a seed (initial value) for a random number generator. During the scrambling process, a newly generated random number is used to scramble each character in a record.

The code phrase is not stored anywhere in the system. Therefore, it is the user's responsibility to remember the code phrase used when scrambling a data set. If the user forgets a code phrase, there is no way the NIH Computer Center can unscramble the data. If the wrong code phrase is used to unscramble a data set, the program will appear to run, but the output data set will be unusable.

The phrase must be at least eight characters long and cannot extend beyond column 72 (any characters in columns 73-80 are ignored). The longer the code phrase, the harder it is for someone to discover it by trying all the possibilities. Since the program converts all lower case characters to upper case, the code phrase can be specified using either lower or upper case characters.

The code phrase consists of the first record of the SYSIN data set. The procedure-stepnames are the same as the procedure names.

3.4.6.1 DSSCM and DSUNSCM

DSSCM creates an output data set, which is the scrambled version of the input data set. DSUNSCM creates an output data set that is the unscrambled version of the input data set. If DSUNSCM is executed on a data set that has not been scrambled, the results will be unpredictable.

Required Symbolic Parameters

NAME Names the input data set.

NEWNAME Names the output data set.

Optional Symbolic Parameters

DISK Supplies the input volume serial number. Required unless the data set is

cataloged.

NEWDISK	Supplies the output volume serial number. Only required if the data set is to be written to a dedicated disk.
STORAGE	Provides the unit name for the input data set; the default is FILE.
NEWSTOR	Provides the unit name for the output data set; the default is FILE.
UNITS	Gives the units for the SPACE parameter; the default is TRK (tracks).
NEWSIZE	Gives the primary and secondary output allocations for the SPACE parameter; the default is '10,10'.
UNUSED	Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.
STEPEND	Gives the disposition for the output data set. The default is KEEP. New data sets on public disks are automatically cataloged.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.

The following example will put a scramble version of the input data set into a TMP data set and catalog the new data set:

Example 31

```
// EXEC DSSCM,NAME='aaaaiii.dsname1',STEPEND=CATLG,  
//      NEWNAME='aaaaiii.dsname2',NEWSTOR=TMP  
//DSSCM.SYSIN DD *  
SECRET CODE PHRASE
```

3.4.6.2 DSSCFI and DSUNSCFI

The DSSCFI and DSUNSCFI procedures update a data set with the scrambled and unscrambled version. If there is a system failure during the scrambling of a data set in place, the contents of a data set will contain a mixture of scrambled and unscrambled records. Since it is extremely difficult to reconstruct a partially scrambled data set, the scrambling in place procedure should be used only if the data can be reconstructed in another way.

Required Symbolic Parameters

NAME Names the input data set.

Optional Symbolic Parameters

DISK Supplies the input volume serial number. Required unless the data set is

cataloged.

STORAGE Provides the unit name for the input data set; the default is FILE.

The following example will replace the scrambled data set with the unscrambled version:

Example 32

```
// EXEC DSUNSCMI,NAME='aaaaiii.dsname'  
//DSUNSCMI.SYSIN DD *  
DON'T FORGET YOUR CODE PHRASE!
```

3.4.7 Transmitting Data to or from Tapes at Remotes

Data transmitted to or from remote workstations by JES2 is either card image or print image. Two procedures, RMTDSIN or RMTDSOUT, are used to transmit tape data containing blocks of different sizes. The RMTDSIN procedure is used when data is to be transmitted from a remote workstation to the central computing facility. The RMTDSOUT procedure is used when data is to be transmitted from the central computing facility to a remote workstation.

The tape transmission schemes supported by the NIH Computer Center have been developed for use with DATA100, Mohawk, and Harris remote workstations. IBM workstations do not support tape transmission. Consult the documentation supplied by the vendor of the particular remote workstation for complete technical information on how to have the workstation initiate transmission of data from the tape drive in the proper format.

Both RMTDSIN and RMTDSOUT return a condition code of zero if there are no errors. Any errors will be listed in the job output and will result in a condition code of four or eight. If the programs invoked by these procedures are unable to write the error message, a condition code of twelve is generated.

These procedures cannot process multiple files in a single execution and are not designed to handle multi-data-set tapes.

The NIH Computer Center does not recommend the use of tape transmissions longer than one hour since it is necessary to restart the transmission from the beginning if the workstation is disconnected for any reason. If it is necessary to disconnect remotes because of problems at the central site, it will be done regardless of tape transmissions in progress at the time.

3.4.7.1 RMTDSIN

RMTDSIN is used to send tape data to the central facility. The program executed by the RMTDSIN procedure will read the data and rewrite it in its original format on a disk or tape. The transmission of data with fixed, variable and undefined length records is supported. With variable length records, RMTDSIN assumes that the correct block and record descriptor words are included in the data transmitted.

Required Symbolic Parameters

NAME Specifies the name of the output data set, which will contain the data transmitted from the remote. A temporary data set would be specified as NAME='&dsname.'

Optional Symbolic Parameters

OPTIONS Specifies which format of the character count record to use. Converting data to and from card images involves the insertion of a character count record before each group of card images representing a block of data on tape. The character count record can be one of three formats. DATA100 is the default. Code MOHAWK or HARRIS for those types of RJE workstations.

STORAGE Specifies the unit name for the output data set (e.g., MSS, SYSDA, TAPE). The default is FILE.

RECFM Specifies the record format of the stored data set. Should be the same as the original format; the default is U.

LRECL Specifies the record length of the stored data set. Should be the same as the original record length; the default is 13030.

BLKSIZE Specifies the blocksize of the stored data set. Should be the same as the original blocksize; the default is 13030.

STATUS Specifies whether the stored data set is new or old; the default is NEW. Code OLD if the data set already exists.

STEPEND Indicates a disposition for output the data set; KEEP is the default. New data sets on public disks are automatically cataloged.

DISK Specifies the volume serial number of the output disk TAPE or tape where the data will be stored. Only required for a data set written to a dedicated disk.

The following four parameters are used only for allocating space for new disk data sets. The default values are sufficient to hold one-to-three million bytes of data on a 3390 disk volume, depending on the blocksize.

UNITS Specifies the units of space to be allocated; the default is TRK for tracks. Code CYL for cylinders or the blocksize for the average block length method.

SIZE	Specifies the number of units in the primary allocation; the default is 10.
INCR	Specifies the number of units in each secondary allocation (up to 15 additional allocations allowed); the default is 5.
UNUSED	Specifies whether any unused tracks should be released or retained; the default is RLSE to release. Code UNUSED= to retain unused space.

The following four parameters are used only when creating a tape data set at the central facility:

DEN	Specifies the density of the tape being created at the NIH Computer Center. DEN should be specified whenever a 9-track tape is accessed.
SEQ	Specifies the data set sequence number of the data set on the tape at the NIH Computer Center. The default for SEQ is 1. Needed only when the data set is the first one on the tape.
LBL	Specifies the type of labels on the tape at the NIH Computer Center. Needed only when the tape does not contain standard IBM labels. Code LBL=NL if the tape is unlabeled.

The RMTDSIN job is read in from magnetic tape data. Then the operator follows the technique defined by the vendor for initiating transmission of data from the tape.

To send magnetic tape data from a Harris remote workstation to the central facility for storage as a cataloged data set in the FILE management class:

Example 33

```
// EXEC RMTDSIN,NAME='aaaaiii.dsname',
//      OPTIONS=HARRIS
//SYSIN DD DATA
      (data from magnetic tape)
```

3.4.7.2 RMTDSOUT

RMTDSOUT is used to transfer data from the central facility to be written on a tape at the remote site. The program executed by RMTDSOUT will generate a data set (SYSOUT Class C) containing the deblocked data, which will be routed to the remote tape unit. Variable length records are transmitted with the descriptor words included.

SYSOUT Class C is used for the purpose of transmitting data to a remote tape unit. The remote tape device should be set to accept only SYSOUT=C data sets. For additional information, refer to the *Remote Job Entry Workstation Guide*.

Required Symbolic Parameters

NAME Specifies the name of the input tape or disk data set to be transmitted to the remote. A temporary data set would be specified as NAME='&dsname'.

Optional Symbolic Parameters

STORAGE Specifies the unit name for the data set to be transmitted (e.g., SYSDA, TAPE). The default is FILE.

OPTIONS Specifies which format of the character count record to use. DATA100 is the default. Code MOHAWK or HARRIS for those types of RJE workstations.

RECFM Specifies the record format of the data set to be transmitted. The default action is to use the RECFM in the data set labels. Therefore, this parameter is needed only for unlabeled tapes, or to override the record format in the data set labels.

LRECL Specifies the record length of the data set to be transmitted. The default action is to use the LRECL in the data set label. Thus, this parameter is needed only for unlabeled tapes or to override the value in the data set labels.

BLKSIZE Specifies the blocksize of the data set to be transmitted to the remote. The default is to use the blocksize from the data set labels. Thus, this parameter is needed only for unlabeled tapes or to override the value in the data set labels.

STEPEND Specifies disposition of the data set to be transmitted; the default is KEEP. New data sets on public disks are automatically cataloged.

DISK Specifies the volume serial number of the tape or TAPE disk containing the data set to be transmitted. Required for a disk data set only if the data set is being stored on a dedicated disk.

The following parameters are used only when transmitting from a tape data set at the central facility:

DEN Specifies the density of the tape being transmitted from the NIH Computer Center. DEN should be specified whenever a 9-track tape is accessed.

SEQ Specifies the data set sequence number of the data set on the tape at the NIH Computer Center. The default for SEQ is 1. Needed only when the data set is not the first one on the tape.

LBL Specifies the type of labels on the tape at the NIH Computer Center. Needed only when the tape does not contain standard IBM labels. Code LBL=NL if the tape is unlabeled.

To receive data for magnetic tape storage at a DATA100 remote workstation from a cataloged data set on an online public disk at the central facility:

Example 34

```
/*ROUTE OUTPUT REMOTEn (if the job is not submitted
    from the remote)
// EXEC RMTDSOUT,NAME='aaaaiii.dsname'
```

3.4.8 ADSMAP for Obtaining Data Set Information

The ADSMAP procedure provides information about a user's data sets. It cannot handle DB2 data sets. For information on DB2 data sets go to:

<http://silk.nih.gov/dbtech>

The FIND DSNAMES control statement is required and needs to be included in the SYSIN data set.

FIND DSNAMES Control Statement

The FIND DSNAMES statement creates two reports, one listing the user's data sets from the nightly index of the public disks (FILE and TMP) and the other for data sets on the MSS. It is similar in function to WYLBUR's SHOW DSNAMES FULL CURRENT command. The listing includes descriptions of each data set including LRECL, BLKSIZE, data set organization, space allocation, key length, and option codes as specified by the DCB parameters. It includes the creation date and date last used for each data set. The estimated daily and monthly storage costs and the total number of tracks allocated for all data sets in the report are also shown. For MSS data sets that have been moved to secondary storage the volume will be reported as MIGRAT and a more limited data set description is provided.

The IN (Indicator) field in the ADSMAP output indicates whether a data set has RACF protection. If the first character of this field is 4, 5, 6, 7, C, D, E, or F, the data set is RACF protected. The IN (Indicator) field does not indicate whether or not a VSAM data set is RACF protected. The AMS LISTCAT command provides this information for VSAM data sets. See the *Using VSAM and AMS at NIH* preface for details.

Example 35

```
//stepname EXEC ADSMAP
//SYSIN DD *
FIND DSNAMES
```

FIND DSNAMES Control Statement Parameters

Four optional parameters may be specified on the FIND DSNAMES statement.

LIKE=aaaaiii.dsname Requests that data sets assigned to the specified account and

initials and beginning with any characters in the dsname be included. Whenever LIKE is included, both the account and initials must be given even if they are the same as those on the JOB statement. If LIKE is not included, the account and initials on the JOB statement will be used.

KEYWORD=kkk	Must be specified on the first request for data sets that have an account/initials pair different from those on the JOB statement or of a preceding request. Subsequent requests for that pair of account/initials do not require the KEYWORD parameter.
DISK	Limits report to public disk volumes (MSS data sets are not included).
TERSE	Requests that empty reports not be printed. For example, if a user has no MSS data sets, no MSS report will be generated.

Rules for Control Statements

- WYLBUR abbreviations are acceptable.
- Commas, blanks, and equal signs are equivalent and may be freely substituted for each other as they can in WYLBUR.
- Columns 73/80 are ignored.
- A maximum of one continuation statement is permitted and is indicated by a comma as the last non-blank character in columns 1/72.
- Up to 100 input statements may be used for a single execution of the procedure.
- Control statements may be in upper case, lower case, or a mixture of the two.

The following example will suppress the printing of empty reports:

Example 36

```
//stepname EXEC ADSMAP  
//SYSIN DD *  
FIND DSNAMES TERSE
```

In the following example, the first control statement will list all data sets beginning with PPPPQQQ (the account and initials on the JOB statement). The second statement will provide a list of the data sets that begin with BBBBJJJ.

Example 37

```
//stepname EXEC ADSMAP  
//SYSIN DD *  
FIND DSNAMES  
FIND DSN LIKE BBBBJJJ KEYWORD=WHA
```

SHOW DSNAMES Control Statement

SHOW DSNAMES has two functions, LIST and MAP. It differs from the FIND DSNAMES control statement in that the information reported is current at the time the procedure is run (rather than being derived from the previous night's index) and only covers cataloged data sets. All catalog entries are listed, even if the data set is not there. Cataloged tape data sets will be included.

Note that the SHOW DSNAMES control statement of ADSMAP incurs substantial overhead due to extensive catalog processing. The FIND DSNAMES control statement is preferable unless users require information about multi-volume tape data sets.

- LIST furnishes the same type of information provided by WYLBUR's SHOW DSNAMES ON CATALOG command and the TSO LISTCAT command. It lists the volume serial number and device type for each data set, in order by data set name. NO LIST may be used to suppress this function.
- MAP provides more detailed information for disk data sets and for primary level MSS data sets; secondary level MSS and tape data sets cannot be mapped. The map provides the same kind of information as WYLBUR's SHOW DSNAMES FULL command, including volume, BLKSIZE, LRECL, DSORG, RECFM, SPACE, and RACF protection information. The map will be in order by volume unless data-set-name order is requested. SH DSNF FULL or MAP should be run in Class E or C.

Requests for mapping information are made from SHOW DSNAMES control statements in the SYSIN data set. The simplest form lists all cataloged data sets and maps the cataloged disk and primary level MSS data sets belonging to the account and initials on the JOB statement. Invalid MSS data sets (e.g., data sets that are not cataloged) will not be listed even though they exist. Examine the JES2 job log of every job that creates MSS data sets for an IEF287I message (Section 4.3) because this is the only practical way to detect such data sets.

The SHOW DSNAMES control statement maps information for non-VSAM data sets only. The Access Method Services (AMS) LISTCAT command will produce the same information for VSAM data sets. See the *Using VSAM and AMS at NIH* preface for details.

Example 38

```
//stepname EXEC ADSMAP
//SYSIN DD *
SHOW DSNAMES MAP
```

It should be noted that RACF currently protects ICF catalog entries for RACF-protected data sets to the same extent that it protects the data sets themselves. This may result in series of messages for RACF-protected data sets of the form:

```
ICH408I JOB(IIIADS ) STEP(ADSMAP )
ICH408I AAAAIII.DATA CL(DATASET ) VOL(DSA111)
```


in the job log of an ADSMAP using the SHOW DSNAMES option (where AAAAIII is the account-initials combination). The data sets referred to in these messages will not be listed or mapped. RACF-registered users may avoid these messages when showing their own RACF-protected data sets by specifying the USER= parameter on the JOB statement of the ADSMAP job. This will also prevent messages for and allow listings and mappings of RACF-protected data sets of others to which the user has at least READ access.

3.4.9 ADSErase to Erase Sensitive Data

ADSErase is a cataloged procedure used to erase a disk data set or an entire tape volume. This procedure is quite inexpensive and should be used to eliminate the possibility of unauthorized access to data protected by the Privacy Act that is no longer needed. It will work on sequential, partitioned, and direct data sets and components of VSAM data sets.

ADSErase will write messages to the SYSPRINT data set indicating what is being erased and whether or not it was successful.

Required Symbolic Parameters

NAME	Specifies the name of the disk data set or the first data set on the tape that will be erased.
------	--

Optional Symbolic Parameters

DISK	Specifies the volume serial number of the disk containing the data set. Required only if the data set is not cataloged.
TAPE	Specifies the volume serial number of the tape containing the data set. Required if the data set is not cataloged. Omitting it will cause the first data set on the tape to be uncataloged.
STORAGE	Specifies the unit name. The default is FILE.
STEPEND	Specifies the final disposition; the default is DELETE.
LABEL	Specifies the label type for tapes; the default is SL.
PARM=IGNDSN	Specifies not to verify the DSN for tapes. When this is included, any data set name can be specified.

The following example shows how to erase a cataloged disk data set and delete the data set at the end of the step.

Example 39

```
//stepname EXEC ADSERASE,NAME='aaaaiii.dsname'
```

The output messages would appear as follows:

```
ADSERASE03 ERASING DISK DATA SET aaaaiii.dsname
           ON VOLUME xxxxxxx
ADSERASE05 ERASING COMPLETE
```

To erase a tape, only the first data set name must be coded. Note that the entire tape will be erased. The following example will erase the data on tape 123456 if the first data set is named AAAAIII.DATA1. (If an unlabeled tape were being erased, any data set name could be specified.)

Example 40

```
//stepname EXEC ADSERASE,TAPE=123456,
//           NAME='AAAAIII.DATA1',
//           STORAGE=TAPE
```

The output messages would appear as follows:

```
ADSERASE02 ERASING TAPE 123456
ADSERASE05 ERASING COMPLETE
```

To use ADSERASE on a VSAM data set, it is necessary to specifically erase each component of the data set separately (although there is probably no need to erase an index component since all it contains is compressed keys). For example, to erase the VSAM KSDS whose cluster name is aaaaiii.VSAM.MAS.BFILE, the following jobsteps must be executed:

Example 41

```
// EXEC ADSERASE,NAME='aaaaiii.VSAM.D.MAS.BFILE',STEPEND=
// EXEC ADSERASE,NAME='aaaaiii.VSAM.I.MAS.BFILE',STEPEND=
```

Both job steps will produce these messages:

```
ADSERASE03 ERASING DISK DATA SET aaaaiii.VSAM.D.MAS.BFILE
           ON VOLUME xxxxxxx

ADSERASE03 ERASING DISK DATA SET aaaaiii.VSAM.I.MAS.BFILE
           ON VOLUME xxxxxxx
ADSERASE05 ERASING COMPLETE
```

STEPEND=DELETE (the default in ADSERASE) is not permitted for a VSAM component and will result in an ABEND if not overridden. Therefore, after the ADSERASE steps are executed, the VDSUTIL utility or AMS DELETE command can be used to delete (scratch) the VSAM cluster. VDSUTIL and AMS DELETE are fully described in the manual *Using VSAM and AMS at NIH*.

3.4.10 EDSUTIL for Data Set Manipulation

EDSUTIL is used for the manipulation of EDIT and non-EDIT format data sets. For example, EDSUTIL can extract portions of files, copy files, and change the DCB characteristics of files while copying. The many capabilities of EDSUTIL and the specifics of EDSUTIL syntax are described in the *Edit Format Utility Package* manual.

This EDSUTIL job copies a data set, changing the DCB characteristics of the output data set.

Example 42

```
// EXEC EDSUTIL
//IN DD DSN=aaaaaaa.dsname1,UNIT=FILE,DISP=SHR
//OUT DD DSN=aaaaaaa.dsname2,UNIT=FILE,
//      DISP=(NEW,CATLG),
//      SPACE=(CYL,(1,1),RLSE),
//      DCB=(RECFM=VB,LRECL=112,BLKSIZE=11200)
//SYSIN DD *
COPY DDNAME=IN TO DDNAME=OUT
```

3.5 TAPE AND TAPE DATA SET UTILITIES

The utilities in this section handle only tapes and tape data sets.

For the most current information on tapes and tape utilities, refer to:

<http://silk.nih.gov/silk/tapes>

3.5.1 Summary of Tape Utilities

Figure 3-4. Overview of Tape Utilities

Procedure Name	Documentation	Function		Symbolic	R/O
RESPOUT	3.5.5	Submit a Micro tape for reprocessing	Pre-assigned Micro tapes	SER DSN	R R
TAPEMAP	3.5.2	Print labels of tape; scan for block count and size of largest block	One tape	TAPE DRIVE DSNUM OPTIONS	R O O O
TAPECOPY	3.5.4	Copy entire tape (all data sets) no editing	Standard-labeled or unlabeled tape	INPUT OUTPUT OPTIONS INUNIT OUTUNIT INDEN OUTDEN	R R O O O O O
PALTAPE	3.5.3 <i>PALTAPE</i>	Analyze, copy, or print a problem tape	One tape with one or more data sets	INTAPE OUTTAPE INUNIT OUTUNIT CORE	R O O O O
TAPESCAN	3.5.7	Add data set information from a tape to the tape management system database	One tape with one or more data sets	TAPE DRIVE	R O
VOLSTAT	3.5.6	Lists status of a tape, date last used and written to.	None	Options	O

Notes:

*R = required; O = optional

**Section numbers refer to Sections in *OS/390 Batch and Utilities*.

Please note that in addition to the utilities listed above, the following utilities, which are described elsewhere in this manual, can also be used with tape data sets:

Procedure	Documentation*	Procedure	Documentation
COPY	3.4.2.1	COMPARE	3.4.5
PRINT	3.4.3.2	SORT	3.4.4.1

* Note: Section numbers refer to Sections in *OS/390 Batch and Utilities*.

The following table may aid the user in deciding which utility to choose depending upon the function to be performed. Where a choice of utilities is given, USE THE FIRST ONE LISTED unless a special feature of the other is required.

Figure 3-5. Functions of Tape Utilities

Functions of Tape Utilities	
Function	Utility
Add data set information from a tape to the tape management system database	TAPESCAN
Change status of a tape - READ or WRITE, PUBLIC or NOPUBLIC	VOLSTAT
Compare two data sets	COMPARE
Copy tape to tape	
SL to SL	
all files, single reel only	TAPECOPY
selected file(s)	DSCOPY, COPY, PALTAPE
SL to NL	DSCOPY, COPY, PALTAPE
NL to SL	DSCOPY, COPY, PALTAPE
NL to NL	
all files, single or multi-reel	TAPECOPY
selected file(s)	DSCOPY, COPY, PALTAPE
AL to SL or NL	DSCOPY, COPY, PALTAPE
Special features for copying:	
Skip to end of file	PALTAPE
Backspace a number of blocks	PALTAPE
Begin at any physical block or logical record	PALTAPE
Reblock	COPY, PALTAPE
Tape with I/O errors	
selected data set(s)	PALTAPE (may specify error option)
all data sets	TAPECOPY (reduced error recovery and/or bypass bad blocks)
Check output tape	TAPECOPY (two methods)
Determine DEN for 9-track	TAPEMAP
Erase data on a tape	ADSERASE
Find out if a tape is labeled	TAPEMAP

Functions of Tape Utilities	
Function	Utility
Print	
Tape labels	TAPEMAP, PALTAPE
Data set labels	TAPEMAP, PALTAPE
Specified file in hexadecimal	PRINT, PALTAPE
Selected records	PRINT, PALTAPE
Specified files	DSLISL, PRINT, PALTAPE
Begin at any record in any block	PALTAPE
Tape with I/O errors	PALTAPE
Number of blocks and largest blocksize for each file	TAPEMAP
Process SL input tape as if NL	PALTAPE
Put tape mark on new NL Special tape or re-label SL Special tape with new VOLSER	Output Distribution Services
Release or reassign a tape	VOLSTAT
Sort/Merge	SORT, MERGE
Update, replace or insert records	IEBUPDTE (tape to tape)

TAPEMAP, TAPECOPY, VOLSTAT, and PALTAPE are NIH-written routines. COPY is the cataloged procedure for IEBGENER; PRINT uses IEBPTPCH; COMPARE uses IEBCOMPR. TAPEMAP, TAPECOPY, and PALTAPE should all be used in a single-step job and should not be mixed with steps using standard data management routines.

PALTAPE is to be used only for tapes that cannot be processed by standard utilities. It is not intended to be a production tool.

Notes on copying tapes:

- No more than two reel tapes may be mounted at one time.
- Each tape requires a /*ROUTE XEQ statement.
- ASCII data may be written only on a user-supplied (special) 9-track tape. (More details are given in *Using Tapes at NIH*.)
- TAPECOPY - all data sets are copied in one step.
- COPY - requires one step per data set to be copied. Cannot copy ASCII variable length data.
- PALTAPE - requires control statements for each data set to be copied.

Other utilities available for saving and retrieving disk or tape data sets are described elsewhere in this section. See the *IBM DFSMS/MVS Utilities* manual, SC26-4926, for IEBUPDTE.

3.5.2 TAPEMAP for Dumping the Labels of a Tape

TAPEMAP is used to display the format of the volume, header, and trailer labels of a standard-labeled or unlabeled tape for one or more data sets on a tape. Appropriate messages are printed for error conditions such as I/O errors, data checks, incorrect density, tape marks embedded in data, and missing labels. The program executes against logical data sets. That is, it treats the header labels, actual data, and trailer labels as a single entity. The first page of TAPEMAP output includes a brief explanation of how labels are formatted, and a map of the volume (VOL1) label of the tape (if the tape is standard-labeled). Because of the way it functions, TAPEMAP does not read the volume label to verify that the correct tape has been mounted; so there is a slight chance that the wrong tape may be mapped.

Each label is printed in two formats. The first describes the different fields of the labels. The second format consists of three print lines of 80 characters each. Line 1 contains the character representation of the label as it exists on tape. Lines 2 and 3 are the hexadecimal equivalent of the character directly above on line 1. This hexadecimal equivalent is printed vertically for ease of association. The first logical data record of each data set requested is printed up to a maximum of 132 characters. It is formatted in three output lines: character data in line 1, and the equivalent hexadecimal representation in lines 2 and 3.

Density information for the labeled 9-track tapes is found in the HDR2 label. For 3480 cartridge tapes, '(N/A)' will be the density field of the HDR2 label. More information on how to determine the density of an unlabeled tape is found in *Using Tapes at NIH*. An example is given later in this section.

TAPEMAP will handle OS, DOS, and ANSI standard-labeled tapes and unlabeled tapes, but will reject non-standard labels. For an ANSI labeled tape, TAPEMAP will indicate whether it is version 1, version 3, or neither. (See *Using Tapes at NIH* for details on accessing ANSI-labeled tapes containing ASCII data.)

If SCAN is specified, the physical block count and the size of the largest block will be printed along with the character representation and hexadecimal equivalent of the first logical data record of each requested data set. For unlabeled ASCII tapes, the character representation will be incorrectly printed.

TAPEMAP can handle only one tape volume at a time. It is not designed to be used as a step within a job that makes other references to the tape. It should be used as a single-step diagnostic tool only.

Required Symbolic Parameters

TAPE Supplies the volume serial number of the tape from which the data is being mapped.

Optional Symbolic Parameters

DRIVE Supplies the unit type of the tape device being used. Defaults to DRIVE=TAPE.

DSNUM Identifies the data sets to be mapped. If several logical data sets are to be mapped, they may be specified in any order. The number of characters within the quotes, including commas, may not exceed 100. The options are:

n Supplies the sequence number of the data set to be mapped. Defaults to DSNUM=1.

'n/m' Denotes a range of data sets (e.g., 2/5 DSNUM='4/LAST')

ALL Maps all data sets on the tape.

LAST Maps the last data set on the tape.

OPTIONS SCAN requests actual block count and size of largest block for each data set being mapped. Defaults to OPTIONS=NOSCAN.

If data checks or equipment checks have occurred, SCAN may be used to see if the data on a tape is usable.

To map the first data set on a cartridge tape:

Example 43

```
//stepname EXEC TAPEMAP,TAPE=tapeser
```

To map and scan all the data sets on a tape:

Example 44

```
//stepname EXEC TAPEMAP,DSNUM=ALL,  
//  OPTIONS=SCAN,TAPE=tapeser
```

To map and scan all data sets on a cartridge tape obtained from another installation:

Example 45

```
//stepname EXEC TAPEMAP,  
//  OPTIONS=SCAN,DRIVE=CTAPE,TAPE=tapeser,DSNUM=ALL
```

3.5.3 PALTAPE for Processing Problem Tapes

PALTAPE is an NIH-written utility to analyze, copy, and print tapes. PALTAPE is to be used only for tapes that cannot be processed by standard utilities. It is not intended to be a production tool.

PALTAPE can perform combinations of the following operations. IEBGENER and IEBPTPCH either cannot provide these facilities or may not be used at the NIH Computer Center for these operations:

- Read a tape and print diagnostic messages specifying type and location of any I/O errors.
- Process tapes with either standard or non-standard labels as if they were unlabeled.
- Skip to end of file.
- Backspace a given number of physical blocks.
- Copy and/or print a data set beginning at any logical record (LRECL) with any physical block (BLKSIZE).
- Process, in one job step, multiple data sets on the same input tape.

In addition, PALTAPE can perform many of the functions provided by IEBPTPCH and IEBGENER. For example:

- Process unlabeled or standard labeled tapes.
- Reblock data on the output tape.
- Add new data sets to an output tape.
- Print data in hexadecimal format.

Restrictions:

- PALTAPE may not be used to create a data set on disk.
- PALTAPE cannot be used to MOD on (i.e., add records) to a data set that already exists on the output tape.
- PALTAPE cannot create more than one output tape data set within one job step. (TAPECOPY should be used to copy several data sets within one job step.)
- PALTAPE can process only one input tape per job step. Therefore, to process a multi-volume data set, process each tape in a separate PALTAPE job step.

- PALTAPE does not process spanned records.

See the separate publication, *PALTAPE—A Utility for Handling Tape Problems*, which is available from the Technical Information Office for guidance in using this utility.

3.5.4 TAPECOPY for Copying Multi-Data-Set Tapes

The TAPECOPY program is used to copy a tape, which contains more than one file. It can be used to copy an OS standard labeled tape to an OS standard labeled tape or to copy an unlabeled tape to an unlabeled tape. (Note however that unlabeled tapes containing imbedded null files cannot be copied beyond the first such file because, in the absence of specific information to the contrary, a null file indicates the end of the data sets on the tape.)

TAPECOPY cannot be used to copy an unlabeled tape to a standard-labeled tape because the file header and trailer labels would be missing. It cannot be used to copy AL labeled tapes except onto a user-supplied (special) tape. This program can be used to copy tapes, which were created by the dump facility of DF/DSS (such as backup tapes created by DISKSAVE).

TAPECOPY does not provide for multi-volume output. It is possible that in some cases (if a tape has been shortened or if an extra-length tape has been brought from another installation), an input volume may contain more information than can be accommodated by the output volume. If a second output volume is required, a message will be issued, and processing will be terminated. It may then be necessary to use the COPY utility, which allows multi-volume output.

When running in a Class C job, TAPECOPY will retain the scratch tape used at the end of each step and assign it to the account/initial combination on the job statement. If more than one step is performed in the job, each tape will be assigned unless a specific VOLSER is specified (e.g., a pre-assigned tape belonging to the user).

Required Symbolic Parameters

INPUT	Specifies the volume serial number of the input tape.
OUTPUT	Specifies the volume serial number of the output tape.

Optional Symbolic Parameters

INUNIT	Specifies the unit name for the input tape. The default is TAPE.
OUTUNIT	Specifies the unit name for the output tape. The default is TAPE.
INDEN	Specifies a code (3 or 4) to indicate the density for OUTDEN the input or output 9-track tape.
OPTIONS	Specifies any of the options listed below.

The options list may include any of the following codes strung together in any order, e.g., OPTIONS=UER. Options R and V are mutually exclusive; the first one requested will be honored.

- B Force bypass of unreadable blocks, logging them on SYSPRINT, the print data set given in the cataloged procedure. (If the tape is really bad, you will probably want to use option 'E' also.)
- E Use reduced error recovery. The number of automatic retries is reduced from 100 to 5.
- V Verify. After copying the tape, read both tapes, performing a byte-by-byte compare to verify the validity of the copy operation. (Note: This option may be costly if it involves a high I/O count.) Use of option V is not allowed with B or E because verification cannot complete successfully if a bad record was bypassed during the copy.
- R After copying the tape, read the copy to see if it is readable. (This costs less than doing a full verify, option V.)
- U Unlabeled tapes will be processed. Both input and output tapes must be unlabeled. When processing an unlabeled tape, the option LABEL=NL is required.

In the following example, the contents of a 9-track 6250 BPI standard-labeled tape are copied onto a cartridge tape.

Example 46

```
//stepname EXEC TAPECOPY,INUNIT=9TRACKHI,INPUT=tapeser,  
//      OUTPUT=tapeser,INDEN=4
```

The following example copies tape 123456 to a SCRATCH tape (running Class C) and assigns it to user AAAAIII (from the JOB statement).

Example 47

```
//IIITAPE JOB (AAAA,box,C), 'JOE PROGRAMMER'  
/*MESSAGE nnnnnn  
/*ROUTE XEQ TAPE  
//STEP EXEC TAPECOPY,INPUT=123456
```

3.5.5 RESPOUT for Reprocessing an Offline Output Tape

The RESPOUT procedure enables the user to produce an additional set of output from an offline output tape (e.g., Micro) if a preassigned tape was used when the tape was created.

Because of the way offline output tapes are handled, the RESPOUT job should not be run until at least 24 hours after the tape is created.

Required Symbolic Parameters

SER Supplies the 6-digit volume serial number of the tape.

DSN Supplies the name of the data set on tape. The name includes a prefix indicating the "type" of output tape to be processed (e.g., MICRO105).

Optional Symbolic Parameter

UNIT Specifies the unit name for the tape. The default is TAPE.

Example 48

```
//STEP1 EXEC RESPOUT,SER=serial,DSN='type.aaaaiii.dsname'
```

To produce another set of output from a multi-volume output data set:

Example 49

```
//STEP1 EXEC RESPOUT,SER='(serial1,serial2)',  
//      DSN='type.aaaaiii.dsname'
```

3.5.6 VOLSTAT for Setting and Showing Volume Status

The VOLSTAT procedure, which is run as a batch job, can be used to list information on each tape volume owned by a user. VOLSTAT creates a status report that lists the volume serial number, the initials and account, the title, and device type for each volume. It also indicates the date on which each volume was issued and last written to, and the type of tape. VOLSTAT also has many useful capabilities to change the attributes of a tape.

No /*MESSAGE statements are needed for VOLSTAT, because the tapes are not mounted.

Information listed by VOLSTAT includes the volume serial number, the owner's account and initials, the title for the volume, the device type, the date issued, the date last used, and the status indicators (e.g., READ ONLY, PUBLIC). Note: for tapes assigned in Class C, the volume title will be the rightmost 17 of the leftmost 34 characters of the DSNNAME parameter.

Control statements for the VOLSTAT program should be supplied following a SYSIN DD * JCL statement. The OPTIONS symbolic parameter can be used to supply a single control statement to the program, and if present, is taken as the first control statement.

To list tapes belonging to the account and initials on the JOB statement, the following JCL is used:

Example 50

```
//STEP1 EXEC VOLSTAT  
//SYSIN DD *  
SHOW TAPES
```

Alternatively the same function can be done using the OPTIONS symbolic parameter:

Example 51

```
//STEP1 EXEC VOLSTAT,OPTIONS='SHOW TAPES'
```

Warning: supplying a keyword on an EXEC statement causes it to be printed in the job output. Use SYSIN for greater security.

A single job step can process a series of control statements. Normally job Class A is sufficient for a VOLSTAT job, however job Class E may be necessary to process a large number of control statements. Selecting specific volume serial numbers is more efficient than requesting all volumes.

Control Statements

VOLSTAT control statements consist of an operation field and one or more operands. Two operations are available, SHOW and SET. Multiple volume serial numbers can be used with either operation. SHOW will display the status of the selected volume(s). LIST is a valid synonym for SHOW. SET will set attributes for selected tape(s) and release or transfer tapes to another user.

A blank may be used instead of the equal sign in the operand field in a control statement. The & symbol is a synonym for AND. Control statements may be up to 72 characters in length. Control statements may be continued onto succeeding lines by placing a comma at the end of the line to be continued. If no ACCESS operand is given, the account and initials from the JOB statement are used to verify volume ownership; if ACCESS is present, the JOB statement information is not referenced.

VOLSTAT control statements will only accept keywords with alphanumeric and national characters.

SHOW Control Statement

The first operand of the SHOW control statement selects the volumes for which status information is to be displayed. This operand is required. Two optional parameters may be used to expand or alter the accounts and initials to be included.

One of the following is required:

TAPES

All tapes will be shown.

```
TAPE=tapeser,  
      (tapeser,tapeser,...)  
      tapeser AND tapeser AND...
```

Selected tape(s) will be shown on the listing.

The optional parameters are:

ACCESS

Volumes belonging to specified owner(s) will be shown. A "/" may be substituted for the "." between the initials and keyword.

```
ACCESS=aaaaiii.kkk,  
      (aaaaiii.kkk,aaaaiii.kkk,...),  
      aaaaiii.kkk AND aaaaiii.kkk AND...
```

ALL

Volumes belonging to all accounts associated with the initials given in the ACCESS parameter or JOB statement will be shown.

The following example will show volume information for all tapes owned under the designated account/initials combination. The account and initials on the JOB statement will not be used. If the keyword given is incorrect, the job will fail.

Example 52

```
//STEP1 EXEC VOLSTAT  
//SYSIN DD *  
SHOW TAPES ACCESS=aaaaiii.kkk
```

Additional examples of SHOW control statements:

- To list all tapes owned by the initials and account on the JOB statement:
SHOW TAPES
- To list all tapes owned by III under any account:
SHOW TAPES ACCESS=AAAIII.KKK ALL
- To list all tapes owned by JJJ or III under the AAAA account:
SHOW TAPES ACCESS=(AAAIII.KKK,AAAJJJ.LLL)
- To list information about tape 012345 owned by the initials on the JOB statement:
SHOW TAPE=012345

Set Control Statement

The SET control statement consists of a tape volume designation, optional access information, and one or more tape attributes to be set.

TAPE=tapeser, Specify selected tape(s) to be set.
(tapeser,tapeser,...)
tapeser AND tapeser AND...

Optional access information may be given:

```
ACCESS=aaaaiii.kkk,
      (aaaaiii.kkk,aaaaiii.kkk,...)
      aaaaiii.kkk AND aaaaiii.kkk AND...
```

TITLE='title'
Specify the tape contents identifier, for example: TITLE=EXPERIMENT '3B'. TTL is an abbreviation for TITLE; FILENAME is a synonym for TITLE. The title may be a maximum of 34 characters.

Mark a tape for release; optionally the user may also indicate that a tape needs to be relabeled (RELABEL) or has other problems (BAD). Note that tapes are not actually released at this time, but an indicator is set. At least once during each business day during working hours, the Information Media Librarian will release all tapes that have the release indicator set. Until the actual release is done, the user may reconsider and decide to reset that indicator to not release the volume. Because the daily tape release may occur early in the business day, tapes released by the user on the last business day of the month may "carry over" and actually be released on the first business day of the next month (resulting in an additional month's charges for the tapes). When the tape is released by the Information Media Librarian, all data sets cataloged to that tape will be automatically uncataloged.

NO RELEASE

Reset the RELEASE indicator so that the volume will be kept, if it has not yet been released. This also clears the RELABEL and BAD indicators. This option does not affect the automatic release of inactive tapes, however the USED option does.

NORELEASE is a synonym for NO RELEASE.

READ

WRITE

Setting READ allows only reading of the tape. Setting WRITE allows reading or writing of the tape.

PUBLIC

NO PUBLIC

Setting PUBLIC allows the tape to be accessed by any job. Setting NO PUBLIC requires that a /*ACCESS statement be included if the account and initials of the tape owner differ from those on the JOB statement of the job trying to access the tape.

NOPUBLIC is a synonym for NO PUBLIC.

USED

Set date-last-used to current date. This causes the tape to be retained in the Information Media Library for another 18 months even if the tape is never mounted. The charge for this option is the same as for mounting a tape.

ASSIGN=aaaaiii

Assign the tape or tapes to a new owner. The external label on the tape volume is changed by the Information Media Librarian to reflect the new ownership. The date last used is not updated.

The following example will cause tape 999999, owned by the initials and account on the JOB statement, to be kept for an additional 18 months.

Example 53

```
//STEP1 EXEC VOLSTAT
//SYSIN DD *
SET TAPE 999999 USED
```

Additional Examples of SET Control Statements

- To set the title field in the tape information:

```
SET TAPE=056789 TITLE='FINAL REPORT'
```

- To reassign a tape from III to JJJ:

```
SET TAPE=222222 ACCESS=AAAAIII.KKK ASSIGN=AAAAJJJ
```

- To mark a tape to be released:

```
SET TAPE 012345 RELEASE
```

- To reset a tape just marked for release so it will not be released:

```
SET TAPE 012345 NO RELEASE
```

- To set a series of specified tapes to READ only status. Continuation on a second line is illustrated:

```
SET TAPE tapeser AND tapeser AND tapeser AND tapeser,  
AND tapeser AND tapeser READ NO DENSITY
```

3.5.7 TAPESCAN

TAPESCAN opens all of the data sets on a single tape so that information can be added to the tape management system database. TAPESCAN cannot be used for tapes that comprise a multi-volume data set.

The following example opens all of the data sets on cartridge tape 012345 and updates the tape management system database:

Example 54

```
//STEP1 EXEC TAPESCAN,TAPE='012345'
```

The following example opens all of the data sets on 9-track tape 123456.

Example 55

```
//STEP1 EXEC TAPESCAN,TAPE=123456,DRIVE=9TRACKHI
```

Required Symbolic Parameters

TAPE	Supplies the volume serial number of the tape from which the data is being scanned
------	--

Optional Symbolic Parameter

DRIVE	Gives a unit name for input; the default is a 3480 cartridge drive.
-------	---

3.6 DISK DATA SET UTILITIES

The utilities described in this section are designed to handle disk data sets. Additional utilities for handling disk data sets may be found in Section 3.4.

3.6.1 Summary of Disk Utilities

Please note that in addition to the utilities listed below, the following, which are described elsewhere in this manual, can also be used with disk data sets:

Procedure	Documentation (Section in this Manual)
ADSERASE	3.4.9
COPY	3.4.2.1
PRINT	3.4.3.2
COMPARE	3.4.5
SORT	3.4.4.1
EDSSORT	3.4.4.2
MERGE	3.4.4.3

Figure 3-6. Disk Utilities

Disk Utilities		
Procedure Name	(Section in this Manual)	Function
ADSRECOV	3.7.4	Recovers a lost data set from a dedicated disk. WYLBUR's ENTER RESTORE command is used to recover data sets to any disk volume.
DISKSAVE	3.7.3	Creates tape backup of a dedicated disk. Maps disk with DISKMAP before copying to tape.
DISKEXPT(IBM utility DF/DSS)	3.7.3.2	Creates tape backup of a dedicated disk for export to another.
DISKGET	3.7.5	Restores complete contents of user's dedicated disk from a backup tape. A DISKMAP is done after the restore operation.
DSSAVE EDSSAVE PDSSAVE DDSSAVE	3.6.2.1 3.6.2.3 3.6.2.5 3.6.2.7	Saves disk data set on tape EDS procedures convert from EDIT format. PDS and DDS procedures unload the data set.
DSGET EDSGET PDSGET DDSGET	3.6.2.2 3.6.2.4 3.6.2.6 3.6.2.8	Copies (reloads) data set from tape to disk. EDS procedures convert to EDIT format.
DISKMAP	3.7.2	Map a disk listing attributes such as primary and secondary space allocations, and number of extents for each data set.
DSRENAME	3.6.3	Renames a data set.
DSSCR	3.6.4.1	Scratches a data set.
DSSCRV	3.6.4.2	Scratches a non-cataloged data set on a specific volume.
DSCOPY	3.4.2.2	Copies a sequential data set from disk to disk or tape to tape.

Disk Utilities		
Procedure Name	Documentation (Section in this Manual)	Function
PDSCOPY	3.6.5.9	Copies a partitioned data set from disk to disk.
DDSCOPY	3.6.7	Copies a direct data set from disk to disk.
PDSALLOC	3.6.5.1	Allocates disk space for a PDS.
MEMADD (IBM program IEBUPDTE)	3.6.5.3	Adds a member to a partitioned data set (source, macro, or private procedure library).
MEMRENAM	3.6.5.4	Renames a member of a PDS.
MEMALTER (IBM program AMASPZAP)	3.6.5.5	Modifies and inspects an existing load module.
PDSDEL	3.6.5.2	Deletes one or more members from a partitioned data set.
PDSMAP	3.6.5.6	Lists member names, space and directory blocks allocated and used.
USRPDSL	3.6.5.7	Lists names only or names and contents of members of a PDS.
PDSPACK (IBM utility IEBCOPY)	3.6.5.8	Condenses a partitioned data set.
USRPDSC	3.6.5.10	Copies one or more members of one partitioned data set to another.
DSLIST	3.4.3.3	Lists the data set.
EDSLIST	3.4.3.4	Lists an EDIT format or sequential data set.
DDSLIST	3.6.6.2	Lists a direct access data set.
PDSLIST	3.6.6.1	Lists contents of members in a PDS.

3.6.2 Saving and Retrieving Disk Data Sets on Tape

The procedures in this section save (copy/unload to tape) and retrieve (copy/reload to disk) data sets. Data sets copied/unloaded to tape via a SAVE utility must be copied/reloaded to disk via the corresponding GET utility. Data set organization determines the correct procedure. The following tables give the data set organization, function, and symbolic

parameters available for each procedure. A detailed explanation of each procedure is given in the section titled by the data set organization (e.g., Sequential Data Sets). These procedures do not support multi-volume data sets. DISKSAVE can be used to back up such data sets; consult TASC when importing a multi-volume data set from another installation.

Figure 3-7. SAVE and GET Procedures

SAVE and GET Procedures				
DATA SET ORGANIZATION	PROCEDURE	FUNCTION	SYMBOLIC PARAMETERS	R/O*
SEQUENTIAL		copying—no conversion for EDIT format save onto tape		
Data set organization is <u>P</u> hysical <u>S</u> equential (PS)	DSSAVE		NAME TAPE DISK NEWNAME STORAGE STATUS DRIVE DEN SEQ LBL STEPEND	R R O O O O O O O O O
	DSGET	retrieve to disk	NAME TAPE DISK NEWNAME STORAGE STATUS DRIVE DEN SEQ UNITS SIZE INCR UNUSED LBL LRECL BLKSIZE RECFM STEPEND	R R O O O O O O O O O O O O O O

SAVE and GET Procedures				
DATA SET ORGANIZATION	PROCEDURE	PROCEDURE FUNCTION	SYMBOLIC PARAMETERS	R/O*
SEQUENTIAL		copying— conversion for EDIT format performed		
Data set organization is <u>P</u> hysical <u>S</u> equential (PS)	EDSSAVE	save onto tape, changing from EDIT to CARD format	NAME TAPE DISK NEWNAME STORAGE STATUS DRIVE DEN SEQ LBL LRECL RECFM BLKSIZE STEPEND OPTIONS UTILOPT	R R O O O O O O O O O O O O O O
	EDSGET	retrieve to disk changing from non-EDIT to EDIT forma	NAME TAPE DISK STORAGE STATUS NEWNAME DRIVE DEN SEQ LBL UNITS SIZE INCR UNUSED LRECL RECFM BLKSIZE STEPEND OPTIONS UTILOPT	R R O O O O O O O O O O O O O O O O

SAVE and GET Procedures				
DATA SET ORGANIZATION	PROCEDURE	PROCEDURE FUNCTION	SYMBOLIC PARAMETERS	R/O*
PARTITIONED				
Data set is <u>P</u> artitioned <u>O</u> rganization (PO)	PDSSAVE	save (unload) onto tape	NAME TAPE DISK NEWNAME STORAGE STATUS DRIVE DEN SEQ LBL STEPEND	R R O O O O O O O O O
	PDSGET	retrieve (reload) to disk	NAME TAPE NEWNAME DISK STORAGE INCR DIR UNUSED DRIVE STATUS DEN SEQ LBL STEPEND	R R O O O O O O O O O O O O
DIRECT				
Data set organization is <u>D</u> irect <u>A</u> ccess (DA)	DDSSAVE	save (unload) onto tape	NAME TAPE DISK NEWNAME STORAGE STATUS DRIVE DEN SEQ LBL	R R O O O O O O O O

SAVE and GET Procedures				
DATA SET ORGANIZATION	PROCEDURE	PROCEDURE FUNCTION	SYMBOLIC PARAMETERS	R/O*
			STEPEND	O
	DDSGET	Retrieve (reload) to disk	NAME TAPE NEWNAME DISK STORAGE STATUS DRIVE DEN SEQ LBL STEPEND	R R O O O O O O O O O

* R = required; O = optional

3.6.2.1 DSSAVE

DSSAVE saves a copy of a sequential disk data set on a tape.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless NEWNAME is specified.

TAPE Supplies the output volume serial number.

Optional Symbolic Parameters

DISK Supplies the input volume serial number. Required only if the data set is not cataloged.

NEWNAME Names the output data set.

STORAGE Provides a unit name for input; the default is FILE.

STATUS Specifies whether the output data set is OLD or NEW; NEW is the default.

DRIVE Gives a unit name for output; the default is TAPE.

DEN Specifies a code (3 or 4) to indicate the density for a 9-track tape.

SEQ Gives the sequence number for the tape data set; the default is 1. Overwriting

data set 1 destroys any subsequent data sets.

LBL Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (non-labeled) may be specified.

STEPEND Indicates the final (end of step) disposition for the output data set; KEEP is the default.

The following example saves a cataloged data set from disk to tape. The data set resides on a disk volume and is being saved to a cartridge tape.

Example 56

```
//stepname EXEC DSSAVE,NAME='aaaaiii.dsname',  
// TAPE=tapeser
```

The following example saves a data set to a cartridge tape.

Example 57

```
//stepname EXEC DSSAVE,NAME='aaaaiii.dsname',  
// TAPE=tapeser
```

The following example saves a cataloged data set from a disk volume to a 1600 BPI tape, makes it the third data set on the tape, and gives it a new name.

Example 58

```
//stepname EXEC DSSAVE,NAME='aaaaiii.dsname1',  
// DRIVE=9TRACKHI,DEN=3,TAPE=tapeser,SEQ=3,  
// NEWNAME='aaaaiii.dsname2'
```

3.6.2.2 DSGET

DSGET copies a tape data set and puts it on a disk.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless NEWNAME is specified.

TAPE Supplies the input volume serial number.

Optional Symbolic Parameters

NEWNAME Names the output data set.

STORAGE Gives a unit name for output; the default is FILE.

STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for input; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1.
UNITS	Gives the units for the SPACE parameter; the default is TRK (tracks).
SIZE	Gives the primary space allocation for the disk data set; the default is 10.
INCR	Gives the number of units in each secondary allocation; the default is 5.
UNUSED	Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.
LBL	Defines the type of label on the tape. SL (standard-labeled) is the default. NL (non-labeled) may be specified, but DCB information must be supplied if the input tape is NL.
LRECL	Gives the logical record length for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.
BLKSIZE	Gives the blocksize for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.
RECFM	Gives the record format for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default. New data sets on public disks are automatically cataloged.
DISK	Supplies the output volume serial number. Required only if the data set is to be written to a dedicated disk.

The following example copies a data set from a cartridge tape to a cataloged data set with a new name in the FILE management class.

Example 59

```
//stepname EXEC DSGET,NAME='aaaaiii.dsname1',
// TAPE=tapeser,NEWNAME='aaaaiii.dsname2'
```

The following example copies a data set from a cartridge tape to a cataloged data set in the MSS management class, changing the default space allocations.

Example 60

```
//stepname EXEC DSGET,NAME='aaaaiii.dsname',  
// TAPE=tapeser,  
// STORAGE=MSS,SIZE=primary,  
// INCR=secondary,UNITS=CYL
```

The following example copies the second data set from an unlabeled 1600 BPI tape to a FILE data set, renaming the data set and cataloging it.

Example 61

```
//stepname EXEC DSGET,NAME='dsname1',  
// TAPE=tapeser,LBL=NL,  
// LRECL=80,BLKSIZE=4000,RECFM=FB,SEQ=2,DRIVE=9TRACKHI,  
// DEN=3,NEWNAME='aaaaiii.dsname2'
```

3.6.2.3 EDSSAVE

EDSSAVE saves a copy of a sequential disk data set on tape and converts the data from EDIT to non-EDIT format. (WYLBUR data sets are automatically saved in EDIT format.)

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is specified.
TAPE	Supplies the output volume serial number.

Optional Symbolic Parameters

DISK	Supplies the input volume serial number. Required only if the data set is not cataloged.
NEWNAME	Names the output data set.
STORAGE	Gives a unit name for input; the default is FILE.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for output; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1. Overwriting data set 1 destroys any subsequent data sets.

LBL	Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified.
LRECL	Specifies a logical record length for the output data set. The default is 80; longer records will be truncated unless LRECL is given.
BLKSIZE	Specifies a block size for the output data set. The default is 11440.
RECFM	Specifies a record format for the output data set. The default is FB.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default.
OPTIONS	The options which may be specified for the output data set are fully described in the <i>Edit Format Utility Package</i> manual. Two of the options are LIST, to create a listing of the input data set and NUMBERED, to place the line numbers in columns 73-80.
UTIOPT	Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the <i>Edit Format Utility Package</i> manual.

The next example will save a cataloged EDIT format data set from a disk volume to tape. Since all defaults are being used, the output data set will be written on a cartridge tape in CARD format.

Example 62

```
//stepname EXEC EDSSAVE,
//  NAME='aaaaiii.dsname',TAPE=tapeser
```

The following example saves a cataloged EDIT format data set from a disk volume to a 6250 BPI tape. Default values for the logical record length and blocksize of the output data set are being overridden.

Example 63

```
//stepname EXEC EDSSAVE,NAME='aaaaiii.dsname',
//  TAPE=tapeser,DEN=4,LRECL=133,BLKSIZE=11438,
//  DRIVE=9TRACKHI
```

The following example saves a cataloged EDIT format data set from a disk volume to a cartridge tape, requesting the placement of line numbers in columns 73 through 80 of each logical record. The output data set will be renamed.

Example 64

```
//stepname EXEC EDSSAVE,NAME='aaaaiii.dsname1',  
// TAPE=tapeser,OPTIONS=NUMBERED,  
// NEWNAME='aaaaiii.dsname2'
```

3.6.2.4 EDSGET

EDSGET copies a tape data set and puts it on a disk. The output data set is converted to EDIT format.

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is specified.
TAPE	Supplies the input volume serial number.

Optional Symbolic Parameters

NEWNAME	Names the output data set.
STORAGE	Gives a unit name for output; the default is FILE.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for input; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1.
UNITS	Gives the units for the SPACE parameter; the default is TRK (tracks).
SIZE	Gives the primary space allocation for the disk data set; the default is 10.
INCR	Gives the number of units in each secondary allocation; the default is 5.
UNUSED	Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.
LBL	Defines the type of label on the tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified, but DCB information must be supplied if the input tape is NL.
LRECL	Gives the logical record length for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.

BLKSIZE	Gives the blocksize for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.
RECFM	Gives the record format for the input data set; it is required if LBL=NL and should not be used for standard-labeled tapes.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default. New data sets on public disks are automatically cataloged.
OPTIONS	The options which may be specified for the input data set are fully described in the <i>Edit Format Utility Package</i> manual. Two of the options are LIST, to create a listing of the input data set and NUMBERED, to obtain the line numbers from columns 73-80 of the input data set.
UTIOPT	Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the <i>Edit Format Utility Package</i> manual.
DISK	Supplies the output volume serial number. Required only if the data set is to be written to a dedicated disk.

The following example retrieves a non-EDIT format data set from a cartridge tape to a disk volume. The output data set will be in EDIT format and will be cataloged.

Example 65

```
//stepname EXEC EDSGET,NAME='aaaaiii.dsname',
// TAPE=tapeser,STORAGE=FILE
```

The following example retrieves a non-EDIT format data set from a 6250 BPI tape received from another installation to a cataloged data set in the FILE management class. The output data set will be in EDIT format and will be renamed and cataloged.

Example 66

```
//stepname EXEC EDSGET,NAME='dsname1',
// TAPE=tapeser,DRIVE=9TRACKHI,DEN=4,
// NEWNAME='aaaaiii.dsname2'
```

The following example retrieves a non-EDIT format data set from a cartridge tape to an MSS cataloged data set. The output data set will be in EDIT format and will be renamed. The default space allocation for the output data set will be overridden; unused space will be retained.

Example 67

```
//stepname EXEC EDSGET,  
// NAME='aaaaiii.dsname1',STORAGE=MSS,  
// TAPE=tapeser,NEWNAME='aaaaiii.dsname2',  
// SIZE=primary,UNUSED=
```

3.6.2.5 PDSSAVE

PDSSAVE unloads a partitioned data set from disk to tape.

Internally, the PDSSAVE procedure invokes the NIH-written program, DSCOPY, which in turn links to the IBM utility program, IEHMOVE.

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is specified.
TAPE	Supplies the output volume serial number.

Optional Symbolic Parameters

DISK	Supplies the input volume serial number. Required only if the data set is not cataloged.
NEWNAME	Names the output data set.
STORAGE	Gives a unit name for input; the default is FILE.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for output; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1. Overwriting data set 1 destroys any subsequent data sets.
LBL	Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default.

The following example saves a cataloged PDS from a disk volume to a cartridge tape.

Example 68

```
//stepname EXEC PDSSAVE,NAME='aaaaiii.dsname',  
// TAPE=tapeser
```

The following example saves a PDS to a 1600 BPI tape and renames it.

Example 69

```
//stepname EXEC PDSSAVE,NAME='aaaaiii.dsname1',  
// TAPE=tapeser,DRIVE=9TRACKHI,DEN=3,  
// NEWNAME='aaaaiii.dsname2'
```

3.6.2.6 PDSGET

PDSGET invokes the IBM utility program IEHMOVE to reload a partitioned data set from tape to disk.

The size of the output data set (primary allocation, directory block, etc.) can be increased when the PDS is reloaded. The size of the output data set is based on the size of the input data set unless a non-default value is specified for one of the symbolic parameters used to determine size.

This procedure has a problem when the output data set is going to a public disk. Unless you override the space request, the step will get a condition code of 8 and you will see the following message in your output "IEH478I MOVE/COPY REQUEST IGNORED BECAUSE TARGET VOLUME(S) MANAGED BY THE STORAGE MANAGEMENT SUBSYSTEM." The cause of the problem is the inability of the IBM utility program IEHMOVE to dynamically allocate new data sets on system managed volumes. To circumvent this problem, override any one of the space parameters (i.e., UNITS=, SIZE=, INCR=, OR DIR=) for the output data set to change it from its default value.

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is specified.
TAPE	Supplies the input volume serial number.

Optional Symbolic Parameters

NEWNAME	Names the output data set.
STORAGE	Gives a unit name for output; the default is FILE.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.

DRIVE	Gives a unit name for input; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1.
LBL	Defines the type of label on the tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified.
UNUSED	Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default. New data sets on public disks are automatically cataloged.
DISK	Supplies the output volume serial number. Required only if the data set is to be written to a dedicated disk.

Note: The size of the output data set is based on the size of the input data set unless a non-default value is specified for one of the following symbolic parameters, which are used to determine size. If a non-default value is given for any of these parameters, the default values will be used for any of the others that are not specified.

UNITS	Gives the units for the SPACE parameter; the default is TRK (tracks).
SIZE	Specifies the number of units in the primary allocation; the default is 10.
INCR	Gives the number of units in each secondary allocation; the default is 5.
DIR	Gives the number of directory blocks to be allocated; the default is 5.
OPTIONS	OPTIONS=NEWSPACE sets the size of the output data set to the default values: UNITS=TRK, SIZE=10, INCR=5, and DIR=5.

The following example retrieves a PDS from a cartridge tape to an MSS data set. The new data set will be cataloged and will be given the same name as the tape data set.

Example 70

```
//stepname EXEC PDSGET,NAME='aaaaiii.dsname',
// STORAGE=MSS,TAPE=tapeser
```

The following example will retrieve a PDS from a cartridge tape received from another installation to a cataloged data set in the FILE management class. The output data set will be renamed.

Example 71

```
//stepname EXEC PDSGET,NAME='dsname1',  
// TAPE=tapeser,DRIVE=CTAPE,NEWNAME='aaaaiii.dsname2'
```

The following example retrieves a PDS from cartridge tape to a cataloged data set in the FILE management class, and renames it. The unloaded PDS is the third data set on the tape.

Example 72

```
//stepname EXEC PDSGET,NAME='aaaaiii.dsname1',  
// NEWNAME='aaaaiii.dsname2',  
// UNITS=CYL,TAPE=tapeser,SEQ=3
```

The following example retrieves a PDS from a cartridge tape to the MSS. Space is allocated for ten directory blocks (to hold approximately 50 entries); unused space will not be released. Because a value is given for DIR, UNITS=CYL must be specified to override the default of UNITS=TRK. The default values of SIZE=10 and INCR=5 will be used.

Example 73

```
//stepname EXEC PDSGET,NAME='aaaaiii.dsname',  
// STORAGE=MSS,  
// UNUSED=,UNITS=CYL,DIR=10,TAPE=tapeser
```

3.6.2.7 DDSSAVE

DDSSAVE unloads a direct access data set from disk to tape.

Internally, the DDSSAVE procedure invokes the NIH-written program, DSCOPY, which in turn links to the IBM utility program, IEHMOVE.

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is specified.
TAPE	Supplies the output volume serial number.

Optional Symbolic Parameters

DISK	Supplies the input volume serial number. Required only if the data set is not cataloged.
NEWNAME	Names the output data set.
STORAGE	Gives a unit name for input; the default is FILE.

STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for output; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1. Overwriting data set 1 destroys any subsequent data sets.
LBL	Defines the type of label on the output tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default.

This example saves a cataloged direct data set from a disk volume to a cartridge tape.

Example 74

```
//stepname EXEC DDSSAVE,
// NAME='aaaaiii.dsname',TAPE=tapeser
```

This example saves a cataloged direct data set from a disk volume as the second data set on a cartridge tape.

Example 75

```
//stepname EXEC DDSSAVE,NAME='aaaaiii.dsname',
// TAPE=tapeser,SEQ=2
```

This example saves a direct data set to a 1600 BPI tape to be sent to another installation.

Example 76

```
//stepname EXEC DDSSAVE,NAME='aaaaiii.dsname',
// DRIVE=9TRACKHI,
// TAPE=tapeser,DEN=3
```

3.6.2.8 DDSGET

DDSGET invokes the IBM utility program IEHMOVE to reload a direct access data set from tape to disk.

This procedure has a problem when the output data set is going to a public disk. Unless you override the space request, the step will get a condition code of 8 and you will see the following message in your output "IEH478I MOVE/COPY REQUEST IGNORED BECAUSE TARGET VOLUME(S) MANAGED BY THE STORAGE MANAGEMENT SUBSYSTEM." The cause of the problem is the inability of the IBM utility program IEHMOVE to dynamically allocate new data sets on system managed volumes. To circumvent this problem, override any one of the space parameters (i.e., UNITS=, SIZE=, INCR=, OR DIR=) for the output data set to change it from its default value.

Required Symbolic Parameters

NAME	Names the input data set; also names the output data set unless NEWNAME is given.
TAPE	Supplies the input volume serial number.

Optional Symbolic Parameters

NEWNAME	Names the output data set.
STORAGE	Gives a unit name for output; the default is FILE.
STATUS	Specifies whether the output data set is OLD or NEW; NEW is the default.
DRIVE	Gives a unit name for input; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.
SEQ	Gives the sequence number for the tape data set; the default is 1.
LBL	Defines the type of label on the tape. SL (standard-labeled) is the default. NL (unlabeled) may be specified.
STEPEND	Indicates the final (end of step) disposition for the output data set; KEEP is the default. New data sets on public disks are automatically cataloged.
DISK	Supplies the output volume serial number. Required only if the data set is to be written to a dedicated disk.

The following parameters are used with a non-specific volume request to select a volume with enough available space to store the data set. Most data sets can be accommodated by the default values. For a very large data set, they may need to be increased.

UNITS Gives the units for the SPACE parameter; the default is TRK (tracks).

SIZE Specifies the number of units in the primary allocation; the default is 10.

This example retrieves a direct data set from a cartridge tape to a cataloged FILE disk data set.

Example 77

```
//stepname EXEC DDSGET,NAME='aaaaiii.dsname',  
// TAPE=tapeser
```

This example retrieves a direct data set from the third data set on a cartridge tape to a data set in the MSS management class.

Example 78

```
//stepname EXEC DDSGET,NAME='aaaaiii.dsname',  
// STORAGE=MSS,UNITS=CYL,TAPE=tapeser,SEQ=3
```

3.6.2.9 Hints on Saving and Retrieving Data Sets

- When exporting an EDIT format data set, use the EDSSAVE procedure. This procedure converts EDIT format data sets to a format that can be read at other installations. On the other hand, when importing a data set that you wish to access through WYLBUR, use the EDSGET procedure. This procedure converts non-EDIT format data sets to EDIT format.
- It is advisable to maintain backups of important and volatile disk data sets rather than relying entirely on the NIH Computer Center's backup facilities.
- Because of the problems that may occur with multi-data set tapes, CIT recommends that no more than 100 data sets be saved per tape.

When changing a data set on a multi-data set tape, all information beyond the point of modification is destroyed. For example, if a tape has 100 data sets and data set 10 is replaced, data sets 11 through 100 are destroyed.

If you are creating a multi-data set tape and have forgotten how many data sets are already on the tape, don't guess. You could destroy important data. Instead, use the TAPEMAP procedure. This procedure gives the data set name and other attributes for each file on the tape.

Be sure to check output before adding more data sets when creating a multi-data set tape. The data sets must be written in direct sequence. If, for example, a previous job to write data set 8 was unsuccessful, the current job to write data set 9 will fail.

- The normal return code from IEHMOVE (used by PDSSAVE and DDSSAVE) is 0. Anything higher should be scrutinized to determine whether the copy was performed.

-
- If the wrong procedure is used (e.g., DDSSAVE for a partitioned data set), the return code will be 8, and the data set will not be copied.
 - The SAVE and GET procedures cannot process data sets where the DSN has a qualifier of more than 8 characters. In this case an error message is issued.

3.6.3 DSRENAME for Renaming Disk Data Sets

The procedure DSRENAME enables users to rename data sets stored on a disk volume. If a data set is referenced through the catalog, the old data set name will be uncataloged and the new data set name will be cataloged. This procedure may be used to rename sequential, partitioned, and direct data sets. It cannot be used for VSAM data sets. See the publication *Using VSAM and AMS at NIH* for further information. For online disk data sets, the TSO and WYLBUR RENAME commands may also be used to perform this function.

DSRENAME cannot receive a data set passed from a previous step.

A step completion code of 000 indicates that the rename operation has completed successfully. Error conditions are indicated by a step completion code of 008 and an explanatory message in the printed output of the DSRENAME step.

Required Symbolic Parameters

NAME Gives the original name of the data set.

NEWNAME Gives the new data set name.

Optional Symbolic Parameters

DISK Supplies the volume serial number of the disk. Required unless the data set is cataloged.

STORAGE Provides a unit name; the default is FILE.

The following example renames and recatalogs a data set.

Example 79

```
//stepname EXEC DSRENAME,NAME='aaaaiii.oldname',  
// NEWNAME='aaaaiii.newname'
```

The following example recalls a large number of staged level 2 data sets. No more than 300 data sets should be recalled in a single batch job. The larger the data sets, the fewer the number that can be recalled in the batch job.

Example 80

```
//stepname EXEC PGM=IEFBR14
//DD1 DD DSN=dsn1,DISP=SHR
//DD2 DD DSN=dsn2,DISP=SHR
//DD3 DD DSN=dsn3,DISP=SHR
.
.
.
//DDn DD DSN=dsn300,DISP=SHR
```

3.6.4 Scratching Disk Data Sets

There are two procedures for scratching disk data sets; DSSCR and DSSCRV.

3.6.4.1 DSSCR for Scratching Data Sets

DSSCR permits a batch job to scratch a single data set stored on a disk. If the data set is cataloged, it will be uncataloged. Even if the data set does not exist, it will be uncataloged.

DSSCR cannot receive a data set passed from a previous step.

Required Symbolic Parameters

NAME Names the data set to be scratched.

Optional Symbolic Parameters

DISK Supplies the volume serial number of a dedicated disk. Required unless the data set is cataloged.

STORAGE Gives the unit name; the default is FILE.

The following example scratches a cataloged data set.

Example 81

```
//stepname EXEC DSSCR,NAME='aaaaiii.dsname'
```

3.6.4.2 DSSCRV for Scratching Non-cataloged Data Sets

DSSCRV scratches data sets on specific disk volumes, without using the catalog to determine the location of the data set. DSSCRV will honor the volume request and scratch the dataset specified in the NAME parameter on the volume specified in the DISK parameter. If the data is cataloged on that volume, it also will be uncataloged. WYLBUR's ENTER SCRATCH command can be used to construct and submit DSSCRV JCL to scratch a dataset on a specific volume.

Required Symbolic Parameters

NAME Names the data set to be scratched.

DISK Supplies the volume serial number of the disk.

The following example scratches a non-cataloged MSS data set from MSSnnn.

Example 82

```
//stepname EXEC DSSCRV,NAME='aaaaiii.dsname',  
// STORAGE=MSS,DISK=MSSnnn
```

The following example scratches a non-cataloged data set on DSA103.

Example 83

```
// EXEC DSSCRV,NAME='aaaaiii.dup.data',DISK=DSA103
```

3.6.5 Creating and Maintaining Partitioned Data Sets

The utilities in this section are used in creating and handling partitioned data sets. The following utilities, which are described elsewhere in this manual, can also be used with partitioned data sets:

Procedure	Section
PDSGET	3.6.2.6
PDSSAVE	3.6.2.5
ADSRECOV	3.7.4
PDSLIST	3.6.6.1

Partitioned data sets can also be created and maintained through WYLBUR. For further information, see the *WYLBUR General Editing* manual.

The following table is designed to help the user choose the appropriate utility or WYLBUR command to use depending on the function to be performed.

Figure 3-8. Functions of PDS Utilities

Functions of PDS Utilities		
	Batch Utility	WYLBUR Command
Reserve space for a PDS	PDSALLOC	CREATE PDS pdsname
List member names and characteristics	PDSMAP	SHOW DSN pdsname FULL SH DSNS pdsname HEX SH MEMBERS OF pdsname SH MEMS LIKE pdsname
List members of PDS (not for load modules)	PDSLIST USRPDSL (not for EDIT-format)	USE FROM pdsname(member-name) then, LIST or LIST FROM pdsname(membername)
List entire PDS	PDSLIST	ENTER PDSLIST
Add/replace a member (not for a load module)	MEMADD, to add MEMADD, to replace	SAVE AS pdsname(membername) RESAVE AS pdsname(membername)
Delete selected members	PDSDEL	SH MEMS OF pdsname SCRATCH SCR pdsname(membername)
Scratch and uncatalog entire PDS	DSSCR	SCRATCH pdsname PDS on CAT
Rename and recatalog entire PDS	DSRENAME	RENAME pdsname AS new-pdsname PDS on CAT
Create a data set from one PDS member	COPY (single PDS member to a new data set)	USE FROM pdsname(membername) SAVE AS data set-name
Reclaim the "dead space"	PDSPACK	CONDENSE pdsname
Reclaim the "dead space" and release unused space	PDSPACK, and code UNUSED=RLSE on EXEC statement	CONDENSE pdsname RELEASE

Functions of PDS Utilities		
Function	Batch Utility	WYLBUR Command
Copy entire PDS	PDSCOPY (new PDS must not exist before running)	none
Copy selected (or all) members to another PDS	USRPDSC (new PDS must exist before running)	USE FROM pdsname(membername) SAVE AS pdsname(membername)
Create a new load module	Any batch job that invokes the Binder program.	none

3.6.5.1 PDSALLOC

PDSALLOC is used to allocate disk space for a partitioned data set on a specific volume. The first step deletes the data set if it exists (if it does not exist, no action is taken); the second step allocates space for the new data set. The stepnames are SCRATCH and ALLOC.

Required Symbolic Parameters

NAME Names the partitioned data set.

Optional Symbolic Parameters

STORAGE Gives a unit name; the default is FILE.

LRECL Specifies a logical record length.

BLKSIZE Specifies a block size.

RECFM Specifies a record format.

UNITS Gives the units for the SPACE parameter; the default is TRK.

SIZE Gives the primary allocation for the SPACE parameter; the default is 20.

INCR Gives the secondary allocation for the SPACE parameter; the default is 10.

INDEX Gives the number of directory blocks for the SPACE parameter; the default is 5. Cannot be increased after the PDS is created. Each block can hold entries for 4 to 5 members.

STEPEND	Indicates a disposition for the data set; KEEP is the default. New data sets on public disks are automatically cataloged.
DISK	Supplies the volume serial number. Required only if the PDS is to be stored on a dedicated disk.

Before running this procedure to an online disk, be sure enough space is available on the disk on which the data set will be stored. (Use WYLBUR's SHOW SPACE command.)

The following example allocates a cataloged partitioned data set in the FILE management class, accepting all defaults:

Example 84

```
//stepname EXEC PDSALLOC,NAME='aaaaiii.dsname'
```

The following example allocates a cataloged partitioned data set in the FILE management class, changing the values for units and primary and secondary space, and giving DCB values:

Example 85

```
//stepname EXEC PDSALLOC,UNITS=CYL,SIZE=10,
// INCR=2,NAME='aaaaiii.dsname',
// LRECL=80,BLKSIZE=11440,RECFM=FB
```

3.6.5.2 PDSDEL

PDSDEL is used to delete one or more members from a partitioned data set. The PDS is not deleted. To delete an entire PDS, use the DSSCR procedure. The space the member took up is not available for re-use until the data set is condensed.

Required Symbolic Parameters

NAME	Names the partitioned data set.
PROGRAM	Names the member(s) to be deleted; cannot exceed 100 characters.

Optional Symbolic Parameters

DISK	Supplies the volume serial number. Required unless the PDS is cataloged.
STORAGE	Gives the unit name; the default is FILE.

The following example deletes a program from a cataloged PDS.

Example 86

```
//stepname EXEC PDSDEL,NAME='aaaaiii.dsname',  
// PROGRAM=progname
```

The following example deletes several programs from a cataloged PDS.

Example 87

```
//stepname EXEC PDSDEL,NAME='aaaaiii.dsname',  
// PROGRAM='progname1,progname2,progname3...'
```

3.6.5.3 MEMADD

MEMADD, which invokes the IBM utility IEBUPDTE, facilitates adding members to source libraries, macro libraries, and private procedure libraries. Note that IEBUPDTE cannot handle members in load module form.

Because JCL conversion (which includes expanding procedures) is done as soon as a job enters the system rather than just before execution, /*BEFORE and /*AFTER cannot be used to separate jobs which will update and then test a procedure in a user's private library. The job to test the procedure must not be submitted until the update job has been run and the new version of the procedure is available.

Required Symbolic Parameters

NAME Names the partitioned data set.

Optional Symbolic Parameters

DISK Supplies the volume serial number of the disk where the PDS is stored. Required only if the PDS is not cataloged.

STORAGE Gives a unit name; the default is FILE.

RECFM Gives the record format of the PDS; required if it is not already defined.

LRECL Gives the logical record length of the PDS; required if it is not already defined.

BLKSIZE Gives the blocksize of the PDS; required if it is not already defined.

OPTIONS Requests any PARM field options for IEBUPDTE. NEW is the default.

CORE Specifies a region size. The default is 4096K.

A SYSIN DD statement must be included to specify the source of input. This would normally be specified as DD * or DD DATA (if the member contains JCL). The input normally

consists of IEBUPDTE ADD statements each of which is followed by the complete text of a member. If //SYSIN DD DATA is used, no /* statements can be included in the data and the input must be terminated with a /* statement. The format of an ADD statement is:

```
. /          ADD    LIST=ALL , NAME=member
```

where 'member' is the name of the member to be added.

The text of the member must immediately follow the ADD statement. If the job is being submitted from WYLBUR, it may be desirable to use a /*UNNUMBERED statement to prevent line numbers from being placed into the PDS. If the member already exists, it will be replaced (unless the OPTIONS parameter is used to specify otherwise). Other IEBUPDTE control statements may be used if desired.

The following example shows how MEMADD may be used to place JCL procedures into an existing cataloged PDS:

Example 88

```
// EXEC  MEMADD , NAME= 'AAAAIII.LIB1 '
//SYSIN DD DATA
. /  ADD  LIST=ALL , NAME=PROC1
      .
      .
      PROG1
      .
      .
. /  ADD  LIST=ALL , NAME=NEWPROC
      .
      .
      NEWPROC
      .
      .
/*
```

3.6.5.4 MEMRENAM

MEMRENAM is used to rename a member of a partitioned data set (PDS) on a disk. Members in PDSs that reside on public disks can be renamed with WYLBUR's RENAME command.

Required Symbolic Parameters

NAME	Complete name of the PDS in which the member to be renamed resides.
MEMBER	1-to-8 character name of the member to be renamed.
NEWMEM	1-to-8 character name of the new member name.

Optional Symbolic Parameters

DISK Volume serial number of the direct access device on which the PDS resides. Required only if the data set is not cataloged.

STORAGE Gives the unit name; FILE is the default.

The following example will rename the member TIME to DATE in the cataloged PDS aaaaiii.ROUTINES.

Example 89

```
//STEP1 EXEC MEMRENAM,NAME='aaaaiii.ROUTINES',  
// MEMBER=TIME,NEWMEM=DATE
```

Members in a PDS can be renamed using the WYLBUR RENAME command.

3.6.5.5 MEMALTER

MEMALTER is provided by the NIH Computer Center only because of the many proprietary software packages being sold to our users which use patching as a maintenance vehicle. This method of maintaining programs is NOT recommended by CIT. It is recommended, instead, that the load module be recreated with the IllvLKMM procedure for the appropriate language.

This procedure, which invokes the IBM utility, AMASPZAP (commonly referred to as "SUPERZAP"), may be used to inspect or modify a load module that is a member of a partitioned data set. The inspection function is controlled by the VERIFY statement. This allows checking the contents of specific locations within the load module. The modification function is controlled by the REP statement.

Because MEMALTER provides the ability to modify programs without the use of proper language translators, misuse of this procedure could result in serious damage to the data set being altered, as well as possible system failure as a result of the modification. The NIH Computer Center will, therefore, record each use of this utility. Remember, formal action will be taken against any individual who attempts to circumvent the protection software provided by the NIH Computer Center and threatens the security of the system.

MEMALTER may be used only with data sets residing on direct access devices that follow the NIH Computer Center standard naming conventions. This includes all data sets on public and private disks.

Required Symbolic Parameters

NAME Names the partitioned data set to be inspected or modified.

Optional Symbolic Parameters

DISK Supplies the volume serial number of the disk. Required only if the PDS is not cataloged.

STORAGE Gives the unit name; FILE is the default. Required only if the PDS is not cataloged.

A //SYSIN DD * statement must be provided to specify the source of the control statements.

The control statements may begin in any column, but the operation function (i.e., NAME, VERIFY or REP) must precede any parameters. There must be at least one blank between the operation and the first parameter. All parameters are specified as multiples of two hexadecimal digits. All control statements are 80-byte records and may not be continued. The control statements are described below:

NAME member csect comment

"member" indicates the name of the load module to be inspected or modified. The load module must be a member of the PDS specified in the above NAME symbolic parameter.

"csect" indicates the name of a particular control section that contains the data to be inspected or modified.

"comment" allows user documentation for this control statement.

The NAME control statement must precede any VERIFY or REP control statements.

VERIFY offset content comment

"offset" provides the hexadecimal displacement of the data to be inspected.

"content" provides the current data of the csect being inspected.

"comment" allows user documentation for this control statement.

REP offset content comment

"offset" provides the hexadecimal displacement of the data to be modified.

"content" provides the new data for the csect being modified.

"comment" allows user documentation for this control statement.

When any modifications are being made, VERIFY statements should precede any REP statements. If the data is not as specified by "content," all REP statements down to the next NAME statement will be ignored.

The following example shows how to replace the characters "BAD" at relative offset 3A with the characters "TRY" in the subroutine HELP contained in the program MYLOAD.

MYLOAD resides in the PDS aaaaiii.PROGRAM.LOADLIB.

Example 90

```
//S1 EXEC MEMALTER,NAME='aaaaiii.PROGRAM.LOADLIB'  
//SYSIN DD *  
NAME MYLOAD HELP  
VERIFY 3A C2C1C4 Character string BAD in HEX  
REP 3A E3D9E8 Character string TRY in HEX
```

3.6.5.6 PDSMAP

PDSMAP lists the names of the members of a partitioned data set in alphabetical order. It also reports the number of tracks and directory blocks allocated to the data set and the number of each used.

Required Symbolic Parameters

NAME Names the partitioned data set to be mapped.

Optional Symbolic Parameters

DISK Supplies the volume serial number. Required only if the PDS is not cataloged.

STORAGE Gives the unit name; FILE is the default.

The following example will map a cataloged PDS.

Example 91

```
//stepname EXEC PDSMAP,NAME='aaaaiii.dsname'
```

3.6.5.7 USRPDSL

USRPDSL permits the user to list the directory or the directory and contents of members in a private library (partitioned data set). USRPDSL can handle non-EDIT format data sets with variable (RECFM=V or RECFM=VB) or fixed length records. The maximum length that will be printed is 121 characters; any record greater than this length will be truncated.

Required Symbolic Parameters

NAME Names the partitioned data set.

Optional Symbolic Parameters

DISK Supplies the volume serial number. Required only if the PDS is not cataloged.

STORAGE Gives the unit name; FILE is the default.

OPTIONS Specifies the type of listing requested. The default is to list only members of the specified library starting with the initials in the jobname of the JOB statement. Two options are available:

- **FULL** causes all members in the library to be referenced, not just those whose names begin with the user's initials.
- **NOLIST** causes only the member's names to be listed instead of the names and contents. If both **FULL** and **NOLIST** are given, **FULL** must come first.

This example will obtain a listing of the contents of each member of the cataloged source library `aaaaiii.MYSORSE` code.

Example 92

```
//stepname EXEC USRPDSL,OPTIONS='FULL',  
// NAME='aaaaiii.MYSORSE'
```

The following example will obtain a listing of just the member names in the source library called `aaaaiii.SRCEPROG`.

Example 93

```
//stepname EXEC USRPDSL,OPTIONS='FULL,NOLIST',  
// NAME='aaaaiii.SRCEPROG'
```

3.6.5.8 PDSPACK

PDSPACK is used to condense a partitioned data set. Condensing is necessary because, otherwise, the space previously occupied by a replaced or deleted member is not available. It is wise to back up the data set on disk (with **PDSCOPY**) or on tape (with **PDSSAVE**) before beginning the **PACK** procedure because, if the **PDSPACK** fails during the condense, the members in the data set are usually destroyed.

Required Symbolic Parameters

NAME Names the partitioned data set to be condensed.

Optional Symbolic Parameters

DISK Supplies the volume serial number. Required only if the PDS is not cataloged.

STORAGE Gives the unit name; the default is **FILE**.

UNITS Gives the units for the SPACE parameter; the default is TRK (tracks).

UNUSED RLSE releases all unused space.

This example condenses a cataloged partitioned data set.

Example 94

```
//stepname EXEC PDSPACK,NAME='aaaaiii.dsname'
```

The UNUSED option releases the unused space and reduces the amount of space for which the user is charged.

This example condenses a cataloged partitioned data set releasing unused space.

Example 95

```
//stepname EXEC PDSPACK,NAME='aaaaiii.dsname',  
//      UNUSED=RLSE
```

If later the user needs additional space for expansion of the PDS, the original secondary space allocation will be used. Note that RLSE releases all unused space, and a data set can end up with less space than was originally allocated in the primary quantity.

This example condenses a PDS, releasing unused space.

Example 96

```
//stepname EXEC PDSPACK,NAME='aaaaiii.dsname',  
//      UNUSED=RLSE
```

3.6.5.9 PDSCOPY

PDSCOPY is a Computer Center procedure that makes it easy to copy an entire PDS from one direct access volume to another. Internally, the PDSCOPY procedure invokes the NIH-written program, DSCOPY, which in turn links to the IBM utility program, IEHMOVE.

The size of the output data set and the size of its directory can be changed when the PDS is copied. The size of the output data set is based on the size of the input data set unless a non-default value is specified for one of the symbolic parameters used to determine size.

This procedure has a problem when the output data set is going to a public disk. Unless you override the space request, the step will get a condition code of 8 and you will see the following message in your output "IEH478I MOVE/COPY REQUEST IGNORED BECAUSE TARGET VOLUME(S) MANAGED BY THE STORAGE MANAGEMENT SUBSYSTEM." The cause of the problem is the inability of the IBM utility program IEHMOVE to dynamically allocate new data sets on system managed volumes. To circumvent this problem, override any one of the space parameters (i.e., UNITS=, SIZE=, INCR=, OR DIR=) for the output data set to change it from its default value

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless NEWNAME is specified.

Optional Symbolic Parameters

INDISK Supplies the input volume serial number. Required only if the data set is not cataloged.

NEWNAME Names the output data set.

INSTOR Provides a unit name for input or output; the default is FILE.
OUTSTOR

UNUSED Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.

STATUS Specifies whether the output data set is OLD or NEW; NEW is the default.

STEPEND Indicates the final (end of step) disposition for the output data set; KEEP is the default. New data sets on public disks are automatically cataloged.

Note: The size of the output data set is based on the size of the input data set unless a non-default value is specified for one of the following symbolic parameters, which are used to determine size. If a non-default value is given for any of these parameters, the default values will be used for any of the others that are not specified.

UNITS Gives the units for the SPACE parameter; the default is TRK (tracks).

SIZE Specifies the number of units in the primary allocation; the default is 10.

INCR Gives the number of units in each secondary allocation; the default is 5.

DIR	Gives the number of directory blocks to be allocated; the default is 5.
OPTIONS	OPTIONS=NEWSPACE sets the size of the output data set to the default values: UNITS=TRK SIZE=10 INCR=5 DIR=5
OUTDISK	Supplies the output volume serial number. Required only if the PDS is to be written to a dedicated volume.

The following example will copy an entire cataloged PDS that is in the FILE management class and place the new, renamed copy in the TMP management class and catalog it. The space allocated for the output data set will be based on the size of the input data set.

Example 97

```
//stepname EXEC PDSCOPY,NAME='aaaaiii.dsname1',
// OUTSTOR=TMP,NEWNAME='aaaaiii.dsname2'
```

In the following example, a cataloged PDS in the FILE management class will be copied and stored in the MSS management class. The default values of INCR=5 and DIR=5 will be used. The new PDS on the MSS will be renamed and cataloged.

Example 98

```
//stepname EXEC PDSCOPY,NAME='aaaaiii.dsname1',
// OUTSTOR=MSS,UNITS=CYL,SIZE=20,
// NEWNAME='aaaaiii.dsname2'
```

3.6.5.10 USRPDSC

USRPDSC allows users to copy one or more members from one library (partitioned data set) to another. The output library must already exist. Members from the input data set will replace any identically-named members in the output data set.

Required Symbolic Parameters

INPUT	Names the input data set from which members will be copied.
OUTPUT	Names the output data set into which members will be copied.

Optional Symbolic Parameters

INDISK	Supplies the input volume serial number. Required only if the PDS is not cataloged.
--------	---

OUTDISK Supplies the output volume serial number. Required only if the PDS is not cataloged.

INUNIT Gives the unit name for the input data set; the default is FILE.

OUTUNIT Gives the unit name for the output data set; the default is FILE.

OPTIONS ALL indicates that all members of the input data set are to be copied.

If **OPTIONS=ALL** is not used, control statements are used to indicate which members are to be copied. The control statements must start in column 1 and contain the names of the members the user wishes to copy; each separated by a comma. Columns 1-66 may be used. Member names may not be continued across control cards, but as many control cards may be used as necessary to contain the list of members.

The following example copies members PROG1, SUB1, SUB2, SUB3, from the cataloged PDS named aaaaiii.TESTLIB into the cataloged PDS named aaaaiii.LOADLIB.

Example 99

```
//stepname EXEC USRPDSC,INPUT='aaaaiii.TESTLIB',  
//      OUTPUT='aaaaiii.LOADLIB'  
//COPY.SYSIN DD *  
           PROG1,SUB1,SUB2,SUB3
```

The following example copies all members from aaaaiii.LIB to the cataloged PDS named aaaaiii.PROGRAMS.

Example 100

```
//stepname EXEC USRPDSC,INPUT='aaaaiii.LIB',  
//      OUTPUT='aaaaiii.PROGRAMS',OPTIONS=ALL
```

3.6.5.11 Other PDS Utilities

Other PDS utilities are described in other sections:

PDSLST	3.6.6.1
PDSGET	3.6.2.6
PDSSAVE	3.6.2.5

3.6.6 Printing Disk Data Sets

The following procedures make it easy to print disk data sets, which have partitioned or direct organization. For further information on listing data sets, see Section 3.4.3.

3.6.6.1 PDSLIST

PDSLIST invokes the LISTPDS program to list the contents of all members or a selected member from a partitioned data set. The data sets can be either EDIT or non-EDIT format. If the data set to be listed has the DCB attributes of RECFM=U and a BLKSIZE greater than or equal to 1022, it is assumed to be an EDIT format data set. The output formats are of two basic kinds: single-line and multi-line.

Each of the single-line formats lists only a single output line corresponding to each input logical record. The use of machine or ASA carriage controls in the input data can be detected by PDSLIST if requested. If this option is not selected, spacing is controlled by the presence or absence of a carriage control indication (machine or ASA) in the data set label or in the JCL. Data with carriage controls (as determined by either of the above methods) will have carriage spacing controlled by these characters; otherwise, single spacing will be used with continued headers between pages.

Each of the multi-line formats lists input records in their entirety by printing the record in 100-character segments. Multi-line formats also have continued headers between pages; double or triple spacing is provided between records; and any carriage controls are listed in the displayed data.

PDSLIST is controlled by the OPTIONS symbolic parameter. Several of the control keywords reference the line number field of input records (defined as the last eight data positions for fixed-format input records and the first eight data positions for U or V-format input records).

A PDS member name may be either an "alias" or a "real" entry. However, since disk storage is associated only with real entries, each alias name should have a corresponding real entry. PDSLIST does not list alias members unless they do not have associated real entries; however, the index entry for each alias entry will be marked as an alias and it will reference the page number of the corresponding real entry.

The WYLBUR ENTER PDSLIST command can also be used to list the contents of all members of an EDIT or non-EDIT format PDS. The PRINT utility described in Section 3.4.3.2 can also be used to list a single member of a PDS.

Required Symbolic Parameters

NAME Names the data set to be listed.

Optional Symbolic Parameters

DISK Specifies the volume serial number of the disk on which the PDS is stored. Required only if the data set is not cataloged.

STORAGE Gives the unit name for the input device; the default is FILE.

DDOUT Supplies printing options (e.g., forms, copies), which will affect only the listing. Gives a code for the SYSOUT=A DD statement to reference a /*DDOUT statement which details the options.

OPTIONS Selects options. Options are listed below under "Option List." If a control keyword is coded more than once, only the last one will affect the program (except that the others must also be syntactically correct). If this symbolic parameter is not coded, then the default options will be FORMAT=LIST, MAX=9999999, PAGE=60, SKIP=60.

Option List

FORMAT=format

Specifies which one of several output formats is desired. The LIST, AORM, SNUM, NUM and NOFORM formats each output a single line for each input record; the MULTI, HEX and DEBUG formats produce as many output lines as required to list each input record fully.

- FORMAT=LIST results in the **LIST**ing and examining of the first record of each member. If the line number field of the first record is numeric, the member will be printed with NUM format; otherwise, NOFORM format will be used.
- FORMAT=AORM determines if both **ASA OR Machine** carriage control is to be used and examines the line number field. Any carriage control indication from the label or the JCL is ignored.
- FORMAT=SNUM Suppresses **NUM**bers (i.e., line numbers) field of each line.
- FORMAT=NUM requests that the line **NUM**ber for each output line be formatted as follows: line number field (with any leading zeroes suppressed), a blank, and the data portion.
- FORMAT=NOFORM requests **NO FORM**atting changes to the data lines being listed.
- FORMAT=MULTI requests **MULTI**-line format, that is, that each input record be printed out fully with 100 characters per output line.

-
- **FORMAT=HEX** requests **HEX**adecimal format, that is, that each input record be printed out fully in a hexadecimal format that uses two output lines for each 100 characters (first line for the upper nibble of each byte—second line for the lower nibble of each byte).
 - **FORMAT=DEBUG** requests that each input record be printed out fully in a combined **MULTI** and **HEX** format.

MAX=number

Specifies the maximum number of input records that are to be processed for any member. For example:

- **MAX=9999999** requests that at most 9,999,999 input logical records be listed for any member.
- **MAX=4** requests that only the first four input logical records be listed for any member.

PAGE=number

Specifies the length (in lines) of an output page. For example:

- **PAGE=60** requests that pages contain no more than 60 lines.
- **PAGE=30** requests that pages contain no more than 30 lines.

SKIP=number

Specifies the number of blank lines that are to be listed between members in the output. Note that upon reaching the top of a following page, no additional blank lines will be produced. For example:

- **SKIP=60** requests that the listing of each member begin on a new page.
- **SKIP=6** requests that six blank lines be printed between members in the output listing.

The syntax scanner that processes the options list is simple. It uses only the first two characters of any keyword and the first two characters of **FORMAT** operands. Commas and/or blanks may be coded as separator characters between the keyword entries. Thus, **OPTIONS='FO=LI,PA=34'** is entirely equivalent to

```
OPTIONS= '    ,FORMAT=LIST    PAGE=34    ' .
```

The following example will list a cataloged PDS.

Example 101

```
//stepname EXEC PDSLIST,NAME='aaaaiii.dsname'
```

The following example will list a single member of a PDS from a cataloged data set using ASA carriage control.

Example 102

```
//stepname EXEC PDSLIST,NAME='aaaaiii.dsname(memname)',  
//      OPTIONS='FORMAT=AORM'
```

3.6.6.2 DDSLIST

DDSLIST lists a direct access data set. If the logical record length is greater than 132, the record will be continued on following lines until it has all been printed.

Required Symbolic Parameters

NAME Names the data set to be listed.

Optional Symbolic Parameters

DISK	Specifies the volume serial number of the disk on which the data set is stored. Required only if the data set is not cataloged.
STORAGE	Gives the unit name for the input device; the default is FILE.
DDOUT	Supplies printing options (e.g., forms, copies), which will affect only the listing. Gives a code for the SYSOUT=A DD statement to reference a /*DDOUT statement which details the options.
OPTIONS	Specifying SUPPRESS suppresses the listing of empty slots (records beginning with hexadecimal FF).

The following example will list a cataloged direct data set stored on a public disk.

Example 103

```
//stepname EXEC DDSLIST,NAME='aaaaiii.dsname'
```

3.6.7 DDSCOPY for Copying a Direct Data Set

DDSCOPY is a Computer Center-written procedure to copy a direct access data set from one direct access device to another. Internally, the DDSCOPY procedure invokes the NIH-written program, DSCOPY, which in turn links to the IBM utility program, IEHMOVE.

This procedure has a problem when the output data set is going to a public disk. Unless you override the space request, the step will get a condition code of 8 and you will see the following message in your output "IEH478I MOVE/COPY REQUEST IGNORED BECAUSE TARGET VOLUME(S) MANAGED BY THE STORAGE MANAGEMENT SUBSYSTEM." The cause of the problem is the inability of the IBM utility program IEHMOVE to dynamically allocate new data sets on system managed volumes. To circumvent this problem, override any one of the space parameters (i.e., UNITS=, SIZE=, INCR=, OR DIR=) for the output data set to change it from its default value.

Required Symbolic Parameters

NAME Names the input data set; also names the output data set unless NEWNAME is specified.

Optional Symbolic Parameters

NEWNAME Names the output data set.

INDISK Supplies the input volume serial number. Required only if the data set is not cataloged.

INSTOR
OUTSTOR Provides a unit name for input or output; the default is FILE.

STATUS Specifies whether the output data set is OLD or NEW; NEW is the default.

STEPEND Indicates the final (end of step) disposition for the output data set; KEEP is the default. New data sets on public disks are automatically cataloged.

OUTDISK Supplies the output volume serial number. Required only if the data set is to be written to a dedicated disk.

The following parameters are used with a non-specific volume request to select a volume with enough available space to store the data set. Most data sets can be accommodated by the default values. For a very large data set, they may need to be increased.

UNITS Gives the units for the SPACE parameter; the default is TRK (tracks).

SIZE Specifies the number of units in the primary allocation; the default is 10.

The following example will copy a direct access TMP data set to a FILE data set, and rename and catalog it.

Example 104

```
//stepname EXEC DDSCOPY,NAME='aaaaiii.dsname1',  
// NEWNAME='aaaaiii.dsname2'
```

3.7 MAINTAINING DEDICATED DISK VOLUMES

Disk volumes reserved for use of a particular user group must be maintained by their owner. Data sets on dedicated disks must be cataloged.

3.7.1 Guidelines for Disk Maintenance

The following suggestions and guidelines are designed to assist users in the proper maintenance of their dedicated disks. Users are responsible for the integrity of the data on all disk volumes that are assigned to them.

3.7.1.1 Technical Coordinator

To avoid possible confusion, a technical person in each group must be appointed to be responsible for the maintenance of the disk volume. This person is responsible for maintaining a current backup of each disk and will be contacted if any disk assigned to that group goes bad and causes problems for the central facility. The technical coordinator and a backup coordinator must be designated at the time the disk volume is requested. The coordinator must be able to contact all users of the disk quickly in case use must be suspended while recovery procedures are performed.

3.7.1.2 Backups

Since the ONLY way to recover a disk volume is by restoring its contents from a good backup tape or, alternatively, recreating every data set on the disk, disks should be dumped (i.e., backed up) to tape FREQUENTLY. The frequency of backups should be determined on the basis of how much activity there is against the disk volume. For example, if a disk is used by many people, is altered every day, or is accessed from TSO or WYLBUR, it should be mapped nightly and backed up at least twice weekly. All of the output should be carefully checked as soon as possible after a maintenance job is run.

Because magnetic tape does deteriorate with time, it is best to have at least four backup tapes, rotating among the tapes for each successive backup job.

If disk backups are not done nightly, active data sets should be backed up individually on some fixed schedule (preferably nightly) if they are vital to a project.

3.7.1.3 DISKMAP for Checking for Missing or Overlapping Tracks

To check for problems with missing or overlapping tracks, the user should execute a DISKMAP of the disk volume either every night or before and after every major run or terminal session modifying vital data sets - whichever occurs **first**.

The DISKMAP output must be checked to verify that the total number of "Tracks Accounted For" (printed at the end of the DISKMAP output) is 39825. If the "Tracks Accounted For" is not correct, check to see whether any tracks are flagged as "Missing," "Overlapping," or "Invalid Extents." IF ONE OF THESE CONDITIONS IS IN EVIDENCE, THE DISK VOLUME WILL HAVE TO BE RE-INITIALIZED. All information stored on the disk is destroyed by the operation.

The STATUS field of the DISKMAP, which is located at the end of the listing, should also be checked. This field indicates whether there was any potential damage to the VTOC and whether it has been corrected.

3.7.1.4 DISKMAP Status Codes

The figure below contains descriptions for the DISKMAP status codes.

Figure 3-9. DISKMAP Status Codes

CODE	INTERPRETATION
X'81' X'89' X'01' X'00'	There is no indication of current or previous damage
X'82' X'02'	An error condition did exist but was corrected automatically by the system
X'04' X'84'	Indicates potential trouble. The next time an attempt is made to allocate space on the disk, the operating system's error correction routines will attempt to correct the error. If they are successful, the field will be reset. If the error correction attempt is not successful, the trouble code will persist. If the error condition does persist, contact TASC.

3.7.1.5 Emergency Procedures

If jobs accessing data sets on a dedicated disk volume ABEND with errors, such as "VTOC ERRORS MAY EXIST" or "DATA CHECKS," or if there are persistent I/O errors on any track on the disk, contact the technical coordinator of the disk immediately.

For any dedicated disk that is suspected of errors, the technical coordinator should see that all users of the disk SUSPEND ALL USAGE OF THE DISK. The coordinator should contact the help desk as soon as possible (by submitting a Service Request Ticket or by telephoning

the TASC consultants). The TASC staff will work with the dedicated disk technical coordinator to ascertain the nature of the problem, to determine the best solution to the problem, and to help the users get started using the disk again.

If the problem with the dedicated disk occurs outside of regular consulting hours and there is a time-critical need to use the disk, the technical coordinator for the dedicated disk can take the following steps:

- Suspend all usage of the disk.
- Call the Computer Operations shift coordinator (301-496-4715), describe the problem and the reasons for "outside of regular hours" action. The Shift Coordinator will work with the disk technical coordinator and will contact Computer Center systems support staff to arrive at the best solution to the problem (for example, initializing a new disk with the volume serial number for the purpose of a disk restore).
- The technical coordinator of the dedicated disk will be contacted by Computer Center staff when the disk problem has been resolved.
- Run the DISKMAP utility on the volume to verify that the correct number of tracks for the entire disk are accounted for.
- Restore the disk from the most recent backup tape. The DISKGET procedure can be used for this purpose.
- Verify that the volume has been restored by checking the output of the DISKMAP portion of the DISKGET job (or by running a separate DISKMAP job).

When the disk has been successfully restored, the technical coordinator should notify users of the disk that usage can resume. Users of the dedicated disk should also be notified whenever any data sets that were updated after the creation of the most recent backup tape will have to be recreated either by re-running the jobs that updated the data sets or by restoring the data sets from their own separate backups.

If Computer Center staff detects hardware errors with a dedicated disk, they will contact the dedicated disk technical coordinator. If necessary, the dedicated disk may be made unavailable to users until the disk coordinator has been contacted.

The procedures for dumping and restoring a dedicated disk volume are documented in Section 3.7.3; the DISKMAP procedure is described in Section 3.7.2.

3.7.1.6 Compressing a Disk

Even though a disk volume may have many tracks of available free space, it is possible that this free space will be "scattered about the disk" in many small pieces. This situation is referred to as fragmentation. When a disk is highly fragmented, the disk coordinator may request a compress of the disk via a Service Request Ticket. A current DISKMAP of the disk must be submitted as documentation for the Service Request Ticket. Before requesting a

compress, the coordinator should insure that the space on the disk is appropriately utilized (e.g., old unused data sets have been deleted, etc.).

The compress is a free service, which is scheduled at an available nighttime/weekend opportunity. It is usually done between midnight and 7:00 a.m. Usage of the volume will have to be suspended for the time frame in which the compress is scheduled. It is advisable to backup the disk immediately after a successful compress.

3.7.1.7 Hints on Maintaining Dedicated Disks

Here are some for maintaining dedicated disks:

- Always check the output of the DISKMAP after each dump and restore job. If you are not executing a procedure, which includes a DISKMAP, include another step in your job to execute DISKMAP before each dump and after each restore.
- Always dump or restore your disks when there is no activity against them (e.g., at night using the LATE resource).
- Batch jobs accessing data sets on a dedicated volume should contain a `/*CNTL controlname,SHR` statement. Backup or restore jobs should contain a `/*CNTL controlname,EXC` statement. In both cases, the "controlname" should be the same.

Note: There is one potential problem that can occur when using `/*CNTL` statements. Because jobs using `/*CNTL` run in strict class order (see Section 2.8.2), if there are enough jobs using SHR and they are higher in the job class order than the backup or restore job, they may prevent the backup or restore job from running during the discount period. To alleviate this, limit the number of jobs to run on days when maintenance is scheduled or require that they use the same or lower job class than the backup or restore job.

If you require help and have followed the guidelines specified above, call TASC and be prepared to deliver a current DISKMAP run of the disk volume to the TASC staff. The consulting staff will attempt to guide and advise you on the best and least expensive way to recover the disk.

For information on maintaining dedicated disks containing RACF-protected data sets, see Section 3.7.2.

3.7.2 DISKMAP for Mapping Private Disks

DISKMAP, which will map the volume table of contents of private disk volumes, is available both as a procedure and as a program which can be executed directly. A report containing information about the success or failure of program execution appears ahead of the listing of data sets.

DISKMAP will produce a listing showing all data sets stored on the specified disk, along with a description of their technical characteristics. Besides indicating the DCB option codes, record format, blocksize, logical record length, key length, and data set organization of each data set, DISKMAP also shows the primary and secondary space allocation and number of extents for each data set on the volume. The space used information is not accurate for VSAM data sets; use the AMS LISTCAT command, instead.) The dates each data set was created (or updated) and last used are given. In addition, a track-by-track map of the volume showing the location of each data set on the volume as well as the location of unused space may be produced.

The IN (INDicator) field in the DISKMAP output indicates whether a data set has RACF protection. If the first character of this field is 4, 5, 6, 7, C, D, E, or F, the data set is RACF protected. The IN (INDicator) field does not indicate whether or not a VSAM data set is RACF protected. The AMS LISTCAT command provides this information for VSAM data sets. See *Using VSAM and AMS at NIH* preface for details.

When executing the DISKMAP procedure, the SYSIN DD statement with one or more MAP control statements may be used to select a variety of options for specified volumes. If only one set of options is needed, the OPTIONS symbolic parameter can perform this same function. The DISKMAP program first checks for the presence of a SYSIN DD statement. If no SYSIN DD statement is present, DISKMAP will look for the PARM field and use it for its command.

The HISTORY DD statement can be used to create a data set containing the DISKMAP-generated records. Additional information about DISKMAP's HISTORY DD statement is provided later in this section.

Details on the use of DISKMAP to detect problems with dedicated disk volumes are given in Section 3.7.1.3.

Required Symbolic Parameters

DISK Supplies the volume serial number.

Optional Symbolic Parameters

STORAGE Gives the unit name; the default is FILE.

CORE Provides a value for region. The default is 2000K.

OPTIONS Selects options. Options are listed below under "Option List."

The following example will list the characteristics of a user's data sets on a private disk.

Example 105

```
//stepname EXEC DISKMAP,DISK=fileser
```

MAP Control Statement

The MAP control statement, which is specified in a SYSIN statement, may be used to select a number of options. The MAP control statement may not extend beyond column 72. Multiple statements may be given.

Syntax of the MAP control statement:

```
MAP (volserlist) optionlist
```

"volserlist" a list of one or more online volumes, separated by commas. If only one volume is specified, the parentheses may be omitted.

"optionlist" contains one or more of the options below, separated by commas. (Optional)

Option List

DSN=NO

Specifies that the listing of data set names on the volume being mapped is not to be printed. The default is DSN=YES.

ISAM=NO

Specifies that the full and detailed description of each ISAM data set appearing in the DSN listing is not to be printed following the DSN listing. The default is ISAM=YES.

TRACKMAP=choice

Specifies whether the map of allocated tracks on the volume is to be printed in standard format by order of track address (TRACKMAP=YES), or data set name (TRACKMAP=BYDSN), or both (TRACKMAP=BOTH), or not printed at all (TRACKMAP=NO). The default is TRACKMAP=YES.

LIKE=aaaaiii/kkk

Specifies that only data sets that begin with aaaaiii as the account and initials are to be printed. "kkk" is the keyword for the aaaaiii combination. Not needed if only data sets beginning with the account and initials on the JOB statement are to be mapped.

The following examples are equivalent:

Example 106

```
// EXEC DISKMAP,DISK=fileser,  
//  OPTIONS='TRACKMAP=NO'
```

Example 107

```
//stepname EXEC DISKMAP  
//DISKMAP.SYSIN DD *  
  MAP fileser,TRACKMAP=NO
```

HISTORY DD Statement

The HISTORY DD statement directs DISKMAP output to a data set, which can then be used as input to other jobs, such as a SAS job. The DD statement, which follows the EXEC statement, should specify a data set name for the report, the volume on which the report should be stored, as well as the DD parameters DISP, UNIT, and SPACE. The output data set created will have DCB characteristics of RECFM=FB, LRECL=137, and BLKSIZE=2740.

The report produced includes the same information as the DISKMAP job output listing that shows all data sets stored on the specified disk along with a description of technical characteristics for each data set. The technical characteristics displayed include DCB option codes, record format, blocksize, logical record length, key length, data set organization, primary and secondary space allocation, the date that the data set was created, and the date that the data set was last used. In addition, the report has the volume serial number on which the data set resides in columns 132 through 137 of each record. See below for the information produced by DISKMAP and the column numbers and field lengths for each field. These field descriptors can be used within a SAS program or any other program written to analyze or reformat the DISKMAP output.

Field	Type	Columns	Length
DSNAME	char	2-45	44
ACCT	char	2-5	4
INITIALS	char	6-8	3
DSORG	char	47-48	2
RECFM	char	50-54	5
BLKSIZE	num	56-60	5
LRECL	num	62-66	5
TRACKS USED	char	68-72	5
TOTAL	num	74-78	5
SECONDARY SIZE	num	80-84	5
SECONDARY TYPE	char	86-88	3
EXTENTS	num	90-91	2
DATE CREATED	char	93-100	8
DATE LAST USED	char	102-109	8

KEY LENGTH	char	112-113	2
DCB OPTION CODE		115-116	2
RACF INDICATOR	char	118-119	2
VOLUME	char	132-137	6

The following example uses the HISTORY feature to collect information from two volumes for one user and save the information concerning the data sets found on these volumes in a new cataloged data set, MAPOUT, in the TMP management class:

Example 108

```
//stepname EXEC DISKMAP
//DISKMAP.HISTORY DD DSN=aaaaiii.MAPOUT,DISP=(,KEEP),
// UNIT=TMP,SPACE=(TRK,(1,1),RLSE)
//DISKMAP.SYSIN DD *
    MAP vol001,LIKE=aaaaiii/kkk
    MAP vol002,LIKE=aaaaiii/kkk
```

Using the DISKMAP Program

The following JCL will map all data sets on the specified disk belonging to the account and initials on the JOB statement. The data set defined by the REPORT DD statement will contain the report concerning run status. However, if there is an error and the REPORT DD statement is missing, the job will fail with a return code of 32. The DASD DD statement can be used to allocate the volume to be mapped instead of using a control statement in the SYSIN DD stream or the PARM field. The DASD DD statement may be replaced with either a SYSIN DD statement containing control statements or by including one such command in the PARM field on the EXEC statement.

Example 109

```
//stepname EXEC PGM=DISKMAP
//REPORT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSUDUMP DD SYSOUT=A
//DASD DD UNIT=FILE,DISP=OLD,VOL=SER=filesesr
```

The following example will perform the same function as the previous example, but omitting the detailed ISAM report.

Example 110

```
//stepname EXEC PGM=DISKMAP,PARM='MAP filesesr,ISAM=NO'
//REPORT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
```

3.7.3 Backing Up Dedicated Disks

Two procedures are available to create a backup of a dedicated (private) disk. Both use the IBM DF/DSS (Data Facility/Data Set Services) program. DISKSAVE should be used unless

the backup tape is to be exported to another installation. The DISKEXPT procedure should be used if the tape is to be exported. By using the ADSRECOV procedure, the backup tape created with either DISKSAVE or DISKEXPT can be used to retrieve individual data sets, (except VSAM data sets). For more information on ADSRECOV, see Section 3.7.4.

The procedures DISKSAVE and DISKGET each execute a DISKMAP to map the disk volume before it is dumped and after it is restored, respectively. The DISKEXPT procedure does not perform this function. To insure the integrity of the data, all activity against a disk must be suspended when it is being dumped to tape or restored from tape.

It is particularly important to check the output of a backup job to ensure that it completed properly. Check the JES2 Job Log, each step accounting block, and the job accounting block for any indication of unsuccessful completion.

When DISKSAVE and DISKGET procedures are used to maintain dedicated disk volumes that contain RACF-protected data sets, the job executing the DISKSAVE procedure must have at least READ access to all protected data sets. The job executing the DISKGET procedure must have ALTER access to all protected data sets. If the RACFID specified on the JOB statement does not give the job proper access authority, the dump or restore will not complete successfully, and the step completion code will be C 8. In addition, a message:

```
ADR402E ... AUTHORIZATION CHECK FAILED FOR ... dsname
```

(where dsname is the name of the data set) will be written to the SYSPRINT data set for each data set that failed RACF access checking.

3.7.3.1 DISKSAVE

Required Symbolic Parameters

DISK	Supplies the volume serial number of the user's dedicated disk to be backed up.
------	---

Optional Symbolic Parameters

TAPE	Supplies the 6-digit volume serial number for the output tape. Several tapes may be specified as a list of volume serial numbers separated by commas and enclosed within parentheses inside single quotation marks.
STORAGE	Gives the unit name for the input: FILE is the default.
DRIVE	Gives the unit name for the output; the default is TAPE.
DEN	Specifies a code (3 or 4) to indicate the density for a 9-track tape.

The following JCL will dump a dedicated disk onto a cartridge tape.

Example 111

```
//stepname EXEC DISKSAVE,DISK=fileser,TAPE=tapeser
```

Information concerning the backup is added to a table which is used by the ADSRECOV procedure. This table contains one record for each disk volume dumped using the DISKSAVE procedure. As indicated in the following table, the amount of information that can be contained in the record for an individual disk volume varies according to the number of tapes required to back up the disk.

Figure 3-10. ADSRECOV Backup Table Capacity

Number of Tapes	Number of Backups Accommodated
one tape	27 backups
two tapes	20 backups
three tapes	16 backups
four tapes	13 backups
five tapes	11 backups
six tapes	10 backups
seven tapes	9 backups
eight tapes	8 backups

If there is not enough space in the record to add information during a backup procedure, all the information related to the oldest backup is removed to create room for the new backup information. If a backup tape is used again for backup, the old table entry is deleted.

The LIST facility of the ADSRECOV procedure can be used to display the contents of this table.

3.7.3.2 DISKEXPT

DISKEXPT creates a tape backup of a disk, which can be exported to another installation. At the receiving installation the contents of the tape can be used to restore to a 3390 disk using the IBM utility program DF/DSS Release 2.3; earlier versions may not work. The symbolic parameters are the same as the ones used with DISKSAVE. Note: The following JCL will create a backup of a private online disk on a 9-track 6250 BPI tape after listing the VTOC of the disk.

Example 112

```
//stepname EXEC DISKMAP,DISK=fileser  
//stepname EXEC DISKEXPT,DISK=fileser,TAPE=tapeser
```

3.7.4 ADSRECOV for Recovering Data Sets from Dedicated Volumes

The ADSRECOV procedure restores a data set from a user's backup of a dedicated disk. The procedure can also restore all data sets from a user's backup tape and either add them to the existing contents of the dedicated disk or replace the contents of the disk with the contents of the backup tape. Only ADSRECOV can be used to restore the contents of a disk from a tape brought in from another installation. Please note, however, that DISKGET (Section 3.7.5) is more efficient than ADSRECOV to replace the entire contents of an NIH disk.

WYLBUR's ENTER RESTORE command, which is described in the *NIH Computer Center User's Guide*, is used to recover backups of data sets, (including VSAM and RACF-protected) stored in the FILE management class.

WYLBUR's ENTER RECOVER command, which uses the ADSRECOV procedure, provides a convenient way to recover data sets (except BDAM, VSAM, and ISAM) from backups of dedicated volumes.

Neither ADSRECOV nor the ENTER RECOVER command can be used to recover individual RACF data sets; however DISKGET can restore an entire dedicated disk including any RACF-protected data sets. For information on RACF, see the *NIH Computer Center User's Guide*.

VSAM data sets cannot be recovered with either ADSRECOV or the ENTER RECOVER command. For further information on VSAM, see the *Using VSAM and AMS at NIH* preface.

Each time a dedicated disk volume is dumped with DISKSAVE, a table containing the dates and times of each backup is updated.

The ADSRECOV job submitted to the system by the user does not perform the actual recovery itself. Instead, this job will create and submit to the system another job that will actually perform the recovery process for the data set. The output of the second job is always sent to OUTPUT HOLD where it can be accessed through WYLBUR or with RHB250 or printed. The output will be purged at the end of the waiting period; see Section 2.11.1 for details on the purge schedule.

The class of the recovery job submitted by ADSRECOV depends on the number of tapes required and the CPU time requested.

The following tables outline the functions of the RECOVER, RECOVER ALL, LIST, and SET control statements.

Figure 3-11. ADSRECOV - RECOVER Statement Functions

ADSRECOV - RECOVER Statement Functions			
RECOVER Functions	RECOVER Options	R/O*	
Causes ADSRECOV to submit a job that restores a data set to disk from a backup tape.	NAME	R	Sequential
	DISK	R	Partitioned
	NEWDISK	O	Direct Access
	NEWNAME	O	Indexed Sequential
	DATE	O	
	STATUS	O	
	ALL	O	
	TAPE	R	(Additional only if volume serial number of a particular tape is specified or if tape is from outside the NIH Computer Center)
	DRIVE	O	
	TAPENAME	O	
	DEVFMT	O	
	UNITS	O	Additional for Sequential
	SIZE	O	
	INCR	O	
	DSORG=PO	R	Additional for Partitioned
	DIR	O	
	UNITS	O	
	SIZE	O	
	INCR	O	
	UNUSED	O	
	DSORG=DA	R	Additional for Direct Access
	DSORG=IS	R	Additional for Indexed Sequential
	SIZE	O	
	INDEX	O	
	OVFLOW	O	

*R = Required; O = Optional

Figure 3-12. ADSRECOV - RECOVER ALL Statement Functions

ADSRECOV - RECOVER ALL Statement Functions			
RECOVER ALL Functions	RECOVER ALL Options	R/O*	Comments
Causes ADSRECOV to submit a job that restores all information from a backup tape to disk, replacing or adding to present contents.	REPLACE or ADD DISK NEWDISK DATE	R R O O	
	TAPE DRIVE TAPENAME DEVFMT	R O O O	(Additional only if volume serial number of a particular tape is specified or if tape is from outside the NIH Computer Center)

*R = Required; O = Optional

Figure 3-13. ADSRECOV - LIST Statement Functions

LIST Functions	LIST Options	R/O*
Lists date and time of backup tapes for a user's private disks.	DISK	O

*R = Required; O = Optional

Figure 3-14. ADSRECOV - SET Statement Functions

SET Functions	SET Options	R/O*
Controls running and output of the recovery job.	REMOTE=n	O
	PRINT=n	O
	NOTIFY	O
	QUICK	O
	DISCOUNT	O
	TIME	O
	ACCESS	O

*R = Required; O = Optional

More than one RECOVER or RECOVER ALL control statement can be included in the same job. If more than one backup is to be accessed, savings can be effected if the data sets from each backup are grouped together.

ADSRECOV has very generous limits on the number of recoveries that can be done; no more than 100 recover statements can be included in one job and no more than 1000 card images of JCL can be generated.

The following JCL can be used to recover a sequential data set from the most recent backup tape and put it back on its original disk using its original name. No data set with that name currently exists on the disk.

Example 113

```
// EXEC ADSRECOV
//SYSIN DD *
RECOVER NAME='aaaaiii.dsname',DISK=fileserv
```

Control Statement Format

Each control statement contains two fields: the operation field, which defines what operation is to be performed, and an operand field, consisting of parameters containing information needed to perform the operation properly. The parameters in the operand field consist of predefined words (keywords), which are usually suffixed with an equal sign (=) followed by information supplied by the user. Parameters may be specified in any order. If the same parameter is specified more than once, the parameter encountered last is used.

The format of the control statements is:

- The operation field must be first and must be followed by at least one space. It may be preceded by one or more spaces.
- The operand field follows the operation field and contains parameters separated by commas. A blank terminates the operand field and causes the remainder of the line to be treated as a comment.
- Columns 73 through 80 are ignored.
- The operand field may be continued on additional lines. Simply break the operand field immediately following any comma before column 72, and resume on the next line. There is a limit of 24 continuations per statement.

LIST Control Statement

The LIST control statement accesses information on the date and time of each backup of users' dedicated disks. Please note, however, that if the user releases a tape that was used as a backup tape, it will still appear in the user's backup table. For this reason, it is better to recycle private backup tapes rather than release them and assign new ones.

Example 114

```
// EXEC ADSRECOV
//SYSIN DD *
LIST DISK=fileserv
```

Backup information for more than one individual disk or type may be requested if the list is enclosed in parentheses or single quotes. The following LIST control statement will list backup information for two dedicated disks.

LIST DISK= 'ABCD01 ,ABCD05 '

RECOVER Control Statement

General Parameters

NAME='aaaaiii.dsname' - Required

Specifies the name of the data set to be recovered.

DISK=filesr - Required

Specifies the volume serial number of the disk that originally contained the data set. No UNIT designation is needed.

STATUS=OLD - Optional

Code if a preallocated data set is to be used. If the data set is a PDS, the recovered PDS will be merged into the existing PDS. Members with names duplicating those already in the PDS will not be copied. If not specified, a new data set will be created.

NEWDISK=filesr - Optional

To recover to a public online disk, code NEWDISK=FILE, TMP, or MSS. The data set will be recovered to a disk of the specified class and cataloged. The volume serial number of a dedicated disk can be specified to receive the recovered data set. If not specified, the disk that contained the data set being recovered is used. No UNIT designation is required.

NEWNAME='aaaaiii.dsname' - Optional

Specifies a new name for the recovered data set. If not specified, the original data set name of the data set being recovered is used.

DATE=mm/dd/yy - Optional

The date, in the format month/day/year, may be either a specific date that a backup was performed (obtained from the backup table), or the date that the data set was last known to be good. In the latter case, the backup copy with the closest date before or equal to the specified date is used. If the DATE parameter is omitted, the most recent backup copy is used. The year portion of the date parameter is assumed to represent the most recent year ending in yy which is less than or equal to the current year.

The following parameters are available for the user who wants to specify that the recovery be done from a specific tape.

TAPE=tapeser - Optional

Specifies the six-digit volume serial number of the backup tape. Multiple tapes may be specified as a list of tape numbers enclosed in single quotation marks or parentheses and separated by commas. Parentheses should be used if the parameter is to be continued on the next line. If more than one tape is specified, the order in which the tapes are specified must be the same order as was specified when the backup was done.

SPECIAL - Optional

This is a sub-parameter of the TAPE parameter. It specifies that the tape containing the backup is a "special" (i.e., non-NIH) tape. The entire parameter should be coded TAPE=(xxxxxx,SPECIAL) for a single-volume backup tape, and TAPE=((xxxxxx,yyyyyy,zzzzzz,...),SPECIAL) for multiple volumes.

ALL - Optional

Causes data sets to be individually copied from the backup tape to the 3390 disk.

DRIVE=unit - Optional

Specifies the unit name (e.g., 9TRACKHI) for a backup tape produced by DISKEXPT or brought in from another installation. If DRIVE is omitted, the procedure will try to obtain a value from the disk backup table. Then, if the tape is not listed in the backup table, TAPE (for a 3480 cartridge tape) is assumed.

TAPENAME='tape.dsname' - Optional

Specifies the data set name on the backup tape. Required for tapes created at another installation; not needed for tapes created at the NIH Computer Center.

DEVFMT=device type - Optional

Specifies the type of disk from which the backup tape was made. The default value is 3390. This parameter is needed only when the TAPE parameter is used and the backup is of a 3380 disk.

The simplest form of the RECOVER control statement is:

```
RECOVER NAME='aaaaiii.dsname',DISK=fileserv
```

If a data set with the same name already exists on the disk, STATUS=OLD must be coded to replace it when the recovery takes place.

```
RECOVER NAME='aaaaiii.dsname',DISK=fileserv,STATUS=OLD
```

A user may also wish to use the NEWNAME and NEWDISK parameters. NEWNAME gives a new name to the data set after it is recovered. NEWDISK specifies that the recovered data set is to be placed on another disk.

Example 115

```
// EXEC ADSRECOV
//SYSIN DD *
RECOVER NAME='aaaaiii.dsname',DISK=fileserv,
        NEWNAME='aaaaiii.dsname',NEWDISK=fileserv
```

If a user tries to recover a data set to a disk that has a data set of the same name as was specified for the recovered data set the recovery will not be performed unless the STATUS parameter is used. Also, the NEWDISK or NEWNAME parameters may be used to avoid this situation.

As a default, the procedure will use the most recent backup tape. The user can specify that an earlier backup is to be used by including the DATE parameter:

```
RECOVER NAME='aaaaiii.dsname',DISK=fileserv,DATE=mm/dd/yy
```

If this control statement were used, the backup done on the specified mm/dd/yy or, if none was done on that date, the backup done on the nearest preceding date would be used.

As an alternative, the user may specify that the recovery be done from a specific tape.

```
RECOVER NAME='aaaaiii.dsname',DISK=fileserv,TAPE=volser
```

To recover data sets from a backup tape produced by DISKEXPT, or brought in from outside the NIH Computer Center, ADSRECOV needs to be told the type of disk from which the backup tape was made. The parameter DEVFMT=device type is used to supply this information. The default for this parameter is 3390; DEVFMT=3380 is only needed when the backup is of a 3380 disk, and the TAPE parameter is used.

Example 116

```
// EXEC ADSRECOV
//SYSIN DD *
RECOVER NAME='aaaaiii.dsname',TAPE=(xxxxxxx,SPECIAL),
        DISK=fileserv,DEVFMT=3380,DRIVE=TAPE
```

General Space Allocation Parameters

Figure 3-15. ADSRECOV Optional Space Parameters

PARAMETER	TYPE OF DATA SET	DEFAULTS	USE
UNITS=value	Sequential Partitioned Indexed Sequential	TRK TRK CYL (cannot be overridden)	Optional Optional
SIZE=n	Sequential Partitioned Indexed Sequential	20 20 1	Optional Optional Optional
INCR=n	Sequential Partitioned Indexed Sequential	10 10 None (not allowed for ISAM data set)	Optional Optional

All the parameters dealing with specifying space allocation are optional. The space allocation parameters are ignored if they are coded in combination with STATUS=OLD.

UNITS=value

Specifies the units that are to be used for the space allocation: code UNITS=TRK, UNITS=CYL, or UNITS=blocksize. Specifying a blocksize means that the system will calculate the number of tracks required using the blocksize of the data set along with the number of blocks the data set will contain. Cannot be used for ISAM data sets.

SIZE=n

Specifies the initial size of the data set. It is either the number of tracks or cylinders that are required to hold the data set, or the number of blocks contained in the data set.

INCR=n

Specifies the amount of secondary space to be acquired if the data set needs more space than was specified in the SIZE parameter.

Other Parameters

Figure 3-16. Other ADSRECOV Parameters

PARAMETER	TYPE OF DATA SET	DEFAULTS	USE
DSORG=PO	Partitioned		Required
DIR=n	Partitioned	5	Optional
UNUSED=RLSE	Partitioned		Optional
DSORG=DA	Direct Access		Required
DSORG=IS	Indexed Sequential		Required
INDEX=value	Indexed Sequential	None	Optional
OVFLOW=value	Indexed Sequential	None	Optional

Partitioned Data Sets

DSORG=PO

Must be used to indicate partitioned organization.

DIR=n

Specifies the number of 256-byte directory blocks. Each directory block can hold entries for 4 to 5 members. The space allocation for the data set must include enough space for the directory.

UNUSED=RLSE

Is used to indicate that unused space is to be released.

Direct Access Data Sets

DSORG=DA

Must be coded to indicate direct access organization. SPACE parameters will be ignored; information is taken from the unloaded tape. Any unused space will be released. If STATUS=OLD is specified, the old data set is scratched and replaced by the new one. Direct access data sets can be recovered with ADSRECOV only to dedicated disks.

ISAM Data Sets

Because ISAM data sets cannot be handled by the single data set restore facility of DF/DSS, the IBM program used by ADSRECOV, the entire disk must be restored before the individual data set is copied. Therefore, the job will be more time consuming and expensive than a typical ADSRECOV job. The NIH Computer Center has discontinued ISAM support; ISAM data sets to be used at the NIH Computer

Center should be converted to VSAM. ISAM data sets can be recovered only to dedicated disks.

DSORG=IS

Must be used to indicate indexed sequential organization.

INDEX=n

Specifies the number of cylinders to be added within the prime area for the index. This number is added to the number of cylinders requested in SIZE; the default is none.

OVFLOW=n

Specifies the number of cylinders for an independent overflow area; the default is none. Note: If not specified, one DD statement is used for allocation; if specified, two DD statements are used.

An ISAM data set is organized differently, depending on the number of DD statements and whether or not the size for the index is specified.

The following information should be useful in determining the organization that ADSRECOV will create:

Number of DD Statements	Index Size Specified	No Index Size Specified
No overflow size specified		
1 DD statement (PRIME)	Prime area with embedded index area. No overflow area.	Prime area with index area at its end. If the unused portion of the index area is less than 1 cylinder, it is used as an overflow area.
Overflow size specified		
2 DD statements (PRIME, OVFLOW)	Separate prime and overflow areas. An index area is embedded in the prime area.	Separate prime and overflow areas. An index area is at the end of the overflow area.

The ADSRECOV procedure will only handle ISAM data sets that exist on one dedicated volume. Users who cannot use this procedure must invoke IEBISAM directly.

RECOVER ALL Control Statement

The RECOVER ALL control statement provides the ability to recover the data sets on a backup tape to a user's private disk.

RECOVER ALL is particularly useful to restore to disk data that was originally on a disk at another installation and was copied to tape for transfer to NIH. A dump tape from another installation may not be restored directly to an NIH disk because changes in VTOC placement would cause severe system problems.

The following parameters are available on the RECOVER ALL control statement.

REPLACE or ADD - Required

Specifies whether the recovered data sets are to replace or be added to the current contents of the disk.

In addition to the above, several parameters of the RECOVER control statement which have been described previously in this section are used with RECOVER ALL. They are:

DISK=fileserv	Required
NEWDISK=fileserv	Optional
DATE=mm/dd/yy	Optional
TAPE=volser	Optional
DRIVE=unit	Optional
TAPENAME='tape.dsname'	Optional

When RECOVER ALL is used, the ADSRECOV procedure will submit a job which will first restore the backup tape to a utility disk and produce a DISKMAP of this utility disk. Next, if REPLACE has been specified, all data sets residing on the user's dedicated disk will be scratched. Finally, the COPY VOLUME function of IEHMOVE is invoked to copy the data sets from the utility disk to the user's dedicated disk. If ADD is specified and an attempt is made to copy a data set whose name already appears on the user's dedicated disk, a message will be issued, the data set will not be copied, and the job will continue with the next data set.

The fact that the recovery job uses IEHMOVE puts certain limitations on what can be copied successfully. ISAM data sets cannot be copied at all, but they may be recovered to a dedicated volume in this or a separate ADSRECOV job by including individual RECOVER control statements for each of them.

The job submitted by an ADSRECOV run that contains RECOVER ALL statements will not run until after 5 p.m.

SET Control Statement

The SET control statement is used to specify characteristics of the recovery job that is submitted by the ADSRECOV procedure. Any number of SET control statements can be

included. If the same parameter appears more than once, the parameter encountered last is used. All the parameters are optional.

```
ACCESS=aaaaiii  
ACCESS=(aaaaiii,aaaaiii, . . .)
```

Specifies the account/initials combination for any tapes to be accessed by the recovery job that are not assigned to the account and initials on the JOB statement. This is never needed when recovering a data set from a public disk volume.

REMOTE=n or PRINT=n

The recovery job is directed to OUTPUT HOLD. This parameter gives a print destination to be used (e.g., REMOTE=99 or PRINT=99) if the output is released from OUTPUT HOLD. If not specified, such released output is printed at the central facility.

NOTIFY

Specifies that notification of job completion for the recovery job will be sent to the user who submitted the ADSRECOV job, if the user is logged onto TSO or WYLBUR when the job completes execution. If not specified, no notification is made to the user when the recovery job completes execution.

QUICK

Specifies that the recovery job is to have the QUICK attribute, for fast processing of jobs submitted between 5:00 p.m. and 7:00 a.m. If not specified, QUICK processing is not performed. Note that unless the ADSRECOV job submits the recovery job during the QUICK processing hours, QUICK will be ignored.

DISCOUNT

Specifies that the recovery job is to be placed in the queue for DISCOUNT processing. If not specified, DISCOUNT processing is not performed unless the job is submitted during the DISCOUNT hours.

TIME=n

Specifies a maximum time for the recovery job in CPU seconds. If it is not specified, the default time for the automatic class of the recovery job (which is based on the number of tapes needed and CPU time) will be used. If TIME specified is more than 75 seconds, the recovery job will be changed to Class C.

The following examples may be helpful in using ADSRECOV.

Example 117

```
// EXEC ADSRECOV
//SYSIN DD *
LIST
RECOVER NAME='AAAAIII.PROG',DISK=ABC002
RECOVER NAME='AAAAIII.DATA',NEWNAME='AAAAIII.INPUT',
        DISK=ABC002,NEWDISK=ABC003
RECOVER NAME='AAAAIII.LETTER',DISK=ABC005,DATE=6/17/99
SET NOTIFY
```

The output of this job will contain a list of dates and times for the backups of the user's dedicated disks. The job will submit a recovery job. The output of the recovery job will be placed in OUTPUT HOLD. The user will receive notification if signed on to WYLBUR or TSO. The most recent backup tape of the user's private disk, ABC002, will be used to recover two data sets; one named AAAAIII.PROG and the other named AAAAIII.DATA, where AAAAIII is the account-initials prefix. The recovered AAAAIII.PROG data set is placed on ABC002 with the same data set name. The recovered AAAAIII.DATA data set is placed on ABC003 with a data set name of AAAAIII.INPUT. The data set named AAAAIII.LETTER will be recovered from the backup of ABC005 created on 6/17/99 or, if none was done on 6/17/99, on the nearest date preceding 6/17/99. The recovered data set is placed on ABC005 using the same data set name.

Example 118

```
// EXEC ADSRECOV
//SYSIN DD *
RECOVER ALL,REPLACE,DISK=ABC005
SET TIME=80
```

This example will cause ADSRECOV to submit a job which will recover all the data sets from the user's most recent backup and restore them to a private disk replacing any data sets in existence on the disk. The recovery job is allotted up to 80 seconds of CPU time which will cause it to run Class C.

3.7.5 DISKGET for Restoring Dedicated Disks

The DISKGET procedure is used to restore the complete contents of a user's dedicated disk from a tape backup. It should not be used to restore a disk from a tape created at another installation (use ADSRECOV) and cannot be used to restore to a different model of disk.

A /*ROUTE XEQ RECOVERY statement, described in Section 2.9.2, is required in every job which uses DISKGET to restore an online disk.

Required Symbolic Parameters

DISK	Supplies the volume serial number of the disk to which the backup tape is being restored.
------	---

TAPE Supplies the volume serial number of the backup tape. Several tapes may be specified as a list of volume serial numbers separated by commas and enclosed within parentheses inside single quotation marks. If more than one tape is specified, they must be specified in the same order used when the backup was performed.

Optional Symbolic Parameters

STORAGE Gives the unit of the disk to which the backup tape is being restored. The default is FILE.

DRIVE Gives the unit name for the tape; the default is TAPE. If the backup tape was created on a 1600 BPI drive, DRIVE=9TRACKHI must be specified.

OLDDISK Supplies the volume serial number of the disk associated with the backup tape. The default is the volume serial number that was specified for the DISK parameter. Do not restore a disk backup containing VSAM data sets to a different volume serial number because the restored VSAM data sets will not be accessible.

The following JCL will restore the contents of a dedicated disk from two 3480 cartridge backup tapes.

Example 119

```
//stepname EXEC DISKGET,DISK=fileser,  
// TAPE='(tapeser1,tapeser2)'
```

3.8 PROCESSING EDIT FORMAT DATA SETS WITH BATCH JOBS

The utilities in this section are designed to convert to and from EDIT format. They each invoke the EDSUTIL program for the conversion. The EDIT format is the standard format used by WYLBUR to store data sets. Each EDIT format line has a number which is stored adjacent to the text of the line. The EDSUTIL program also permits manipulation of data sets from the batch without conversion from EDIT format. It is described in the *Edit Format Utility Package* manual. EDIT format is typically two or three times as efficient in the use of disk space as other formats.

NOTE: EDIT format data sets are not compatible with SILK (Secure Internet LinKed) technology that allows users to display OS/390 data sets through the World Wide Web. For information on SILK, visit:

<http://silk.nih.gov>

3.8.1 Summary of Utilities for EDIT Format Data Sets

Figure 3-17. Utilities for EDIT Format Data Sets

Utilities for EDIT Format Data Sets					
Procedure Name	Documentation	Function	Input	Symbolic Parameters	R/O*
EDSIN	3.8.2.1	Produces non-EDIT format temporary data set which can be read by a batch job	EDIT format data set	NAME DISK STORAGE LRECL RECFM BLKSIZE OPTIONS INPUT SIZE UTILOPT	R O O O O O O O O O O
EDSOUT	3.8.2.2	Converts output data set from a batch job to EDIT format	temporary non-EDIT format data set named &OUTPUT	NAME DISK UNITS SIZE INCR UNUSED STATUS OUTPUT DISP STEPEND UTILOPT	R O O O O O O O O O O
EDSSORT	3.4.4.2	Sorts data set	EDIT format data set	NAME DISK NEWNAME NEWDISK STATUS SIZE STORAGE NEWSTOR UNITS SIZE INCR UNUSED LRECL BLKSIZE UTILOPT	R O R O O O O O O O O O O O O

Utilities for EDIT Format Data Sets					
Procedure Name	Documentation	Function	Input	Symbolic Parameters	R/O*
EDSLIST	3.4.3.4	Lists data set; equivalent to WYLBUR's LIST UNNUMBERED command	EDIT format data set	NAME DISK TAPE STORAGE DEN SEQ LBL LRECL RECFM BLKSIZE DDOUT OPTIONS UTILOPT	R O O O O O O O O O O O O
EDSSAVE	3.6.2.3	Produces non-EDIT format data set on tape	EDIT format data set on DISK	NAME DISK TAPE NEWNAME STORAGE STATUS DRIVE DEN SEQ LBL RECFM LRECL BLKSIZE STEPEND PTIONS UTILOPT	R O R O O O O O O O O O O O O O
EDSGET	3.6.2.2	Produces non-EDIT format data set on disk	non-EDIT format data set on tape	NAME DISK TAPE STORAGE DRIVE NEWNAME STATUS DEN SEQ LBL	R O R O O O O O O O

Utilities for EDIT Format Data Sets					
Procedure Name	Documentation	Function	Input	Symbolic Parameters	R/O*
				RECFM	O
				LRECL	O
				BLKSIZE	O
				UNITS	O
				SIZE	O
				INCR	O
				UNUSED	O
				STEPEND	O
				OPTIONS	O
				UTILOPT	O

Note: *R = required; O = optional

3.8.2 Reading and Writing EDIT Format Data Sets

Procedures have been developed which allow the WYLBUR user to manipulate data sets stored in EDIT format, the normal form for a WYLBUR data set. Programmers should use these procedures and not try to write their own to convert to and from EDIT format. CIT will not pre-announce internal changes to the EDIT format software, and assumes no responsibility if changes to this software cause a job that invokes it illegally to fail. As an example, the internal format of WYLBUR's EDIT format data sets is subject to change without advance notice.

Procedures to sort (EDSSORT), list (EDSLIST), and store and retrieve data sets on tape in EDIT format are described in other areas of this manual.

None of the EDS procedures are designed for use with members of a PDS.

3.8.2.1 EDSIN

EDSIN allows a batch job to read an EDIT format data set. It converts the data set to fixed-blocked format and stores it into a temporary data set named &INPUT which is PASSED to a later job step.

Required Symbolic Parameters

NAME Specifies the name of the EDIT format data set that will be input to the program.

Optional Symbolic Parameters

DISK	Specifies the volume serial number of the disk where the EDIT format data set is stored. Required only if the data set is not cataloged.
LRECL	Specifies the logical record length for the temporary data set. Cannot exceed 500. LRECL=80 is the default.
RECFM	Specifies the record format for the temporary data set. RECFM=FB is the default.
BLKSIZE	Specifies the blocksize for the temporary data set. It must be a multiple of the LRECL. BLKSIZE=11440 is the default.
STORAGE	Gives a unit name for input; the default is FILE.
INPUT	Specifies a name for the temporary input data set. INPUT='&INPUT' is the default.
OPTIONS	The options which may be specified for a temporary data set are fully described in the <i>Edit Format Utility Package</i> manual. Two of the options are LIST, to create a listing of the input data set and NUMBERED, to place the line numbers in columns 73-80.
UTOLOPT	Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the <i>Edit Format Utility Package</i> manual.
SIZE	Specifies a primary space allocation for output. The default is 5 cylinders.

For example, the following job control language statements can be used to compile a program that is stored as a cataloged WYLBUR data set:

Example 120

```
//stepname EXEC EDSIN,NAME='aaaaiii.dsname '  
//stepname EXEC lllvCOMP  
//COMP.SYSIN DD DSN=&INPUT,DISP=(OLD,DELETE)
```

where "lll" is the language prefix
"v" is the version (e.g., FORVCOMP for VS FORTRAN)

The following job control language permits a user's program to read data from a cataloged EDIT format data set:

Example 121

```
//stepname EXEC EDSIN,NAME='aaaaiii.dsname '  
//stepname EXEC lllvLKGO  
//GO.SYSIN DD DSN=&INPUT,DISP=(OLD,DELETE)
```

where "lllvLKGO" is the name of the procedure for the bind and run step (e.g., ASMFLKGO).

EDSIN normally changes the data set to card images. To change it to records of some other length, use:

Example 122

```
//stepname EXEC EDSIN,NAME='aaaaiii.dsname',  
// LRECL=record,BLKSIZE=block
```

When converting an EDIT format data set to card images, the line numbers can be placed in columns 73 through 80:

Example 123

```
//stepname EXEC EDSIN,NAME='aaaaiii.dsname',  
// OPTIONS=NUMBERED
```

A temporary name other than &INPUT can be used.

Example 124

```
//stepname EXEC EDSIN,NAME='aaaaiii.dsname',  
// INPUT='&temp'
```

where "&temp" is the temporary name to be used. Remember that it must be 8 or fewer characters and must start with an ampersand.

This is useful if a job step is to read several EDIT format data sets since each data set must have a different dsname.

3.8.2.2 EDSOUT

EDSOUT is used to convert output from a batch job to EDIT format. It assumes that the output is in a temporary data set named &OUTPUT which is PASSED from an earlier job step. The logical record length of a data set which is input to EDSOUT should not exceed 500 (504 for Variable format) and the blocksize should not exceed 19,116.

Required Symbolic Parameters

NAME	Specifies the name of the output EDIT format data set which will be stored on disk.
------	---

Optional Symbolic Parameters

DISK	Specifies the volume serial number of the disk where the EDIT format data set is to be stored. Required only if the data set is not cataloged.
STORAGE	Specifies the unit name for output; the default is FILE.
STATUS	OLD indicates that a data set with the same name is to be replaced. NEW is the default.
UNITS	Gives the units for the SPACE parameter; the default is TRK (tracks).
SIZE	Gives the primary allocation for the SPACE parameter; the default is 10.
INCR	Gives the number of units in each secondary allocation; the default is 5.
UNUSED	Determines whether unused space is released; the default is RLSE. Code UNUSED= to retain unused space.
OUTPUT	Specifies a name for the temporary output data set. OUTPUT='&OUTPUT' is the default.
DISP	PASS specifies that the temporary data set is to be used in subsequent steps. DELETE is the default.
OPTIONS	The options which may be specified for the input data set are fully described in the <i>Edit Format Utility Package</i> manual. Two of the options are LIST, to create a listing of the input data set and NUMBERED, to obtain the line numbers from columns 73-80 of the input data set.
STEPEND	Indicates the final (end of step) disposition for the output data set. KEEP is the default. New data sets on public disks are automatically cataloged.
UTOPT	Specifies parameters passed to the EDSUTIL program to affect message and ABEND processing. For further information, consult the <i>Edit Format Utility Package</i> manual.

For example, to convert normal printer output from a FORTRAN program and save it as a cataloged EDIT format data set in the FILE management class, use the following job control language:

Example 125

```
//stepname EXEC FORVCOMP
//COMP.SYSIN DD *
(FORTRAN program)
/*
//stepname EXEC FORVLKGO
//GO.FT10F001 DD DSN=&OUTPUT,UNIT=SYSDA,
// SPACE=(TRK,(30,10),RLSE),DISP=(,PASS),
// DCB=(RECFM=FB,LRECL=133,BLKSIZE=11438)
//stepname EXEC EDSOUT,NAME='aaaaiii.dsname'
```

A data set that already exists can be overwritten:

Example 126

```
//stepname EXEC EDSOUT,NAME='aaaaiii.dsname',
// STATUS=OLD
```

A different temporary data set name can be specified:

Example 127

```
//stepname EXEC EDSOUT,NAME='aaaaiii.dsname',
// OUTPUT='&temp'
```

where "&temp" is the temporary name to be used. Remember that it must be 8 or fewer characters and must start with an ampersand.

This is useful if a job step has several output data sets that are to be converted to EDIT format since each one must have a different dsname.

The default step-end disposition is DELETE for the input data set. If the data set is to be used in subsequent steps, DISP=PASS should be coded.

Example 128

```
// EXEC EDSOUT,NAME='aaaaiii.dsname',DISP=PASS
```

Storing GO Step Output into a WYLBUR Data Set (EDSOUT)

The following general example of job control language statements can be used to store GO step output into a cataloged WYLBUR data set:

Example 129

```
//stepname EXEC lllvLKGO
//GO.ddname DD UNIT=SYSDA,DISP=(NEW,PASS),
// DCB=(RECFM=format,LRECL=record,BLKSIZE=block),
// SPACE=(TRK,(10,5),RLSE),DSN=&OUTPUT
//GO.SYSIN DD *
(data)
//stepname EXEC EDSOUT,NAME='aaaaiii.dsname'
```

where "lllvLKGO" is the name of the bind and GO procedure for the language being used (e.g., FORGLKGO, CBLALKGO).

"ddname" is the name defined by the program or by the language for the output data set.

"format" is the format of the created data set, including the kind of carriage control which is either defined in the program or defined by the language.

"record" is the number of characters per line of output plus 1 for a carriage control character, and can be changed by the programmer; it must be less than or equal to 500.

"block" is an integral multiple of "record." The largest value that does not exceed 6356 is recommended.

Specific information about the "standard" output formats may be found in the appropriate language section.

3.9 USING THE BINDER

The IBM Program Manager Binder is a service program used to prepare executable programs from object modules. There are three steps in the process of creating an executable program. The first step is to write a program in a source language such as COBOL, FORTRAN, or PL/I. Next, the source program is translated by a compiler into an object module. Finally, the object module is fully resolved by the Binder to load modules and/or other object modules to create an executable load module.

Load modules can be either fully or partially resolved. Fully resolved means that all the programs necessary to execute are stored in the load module. Partially resolved means that some of the programs necessary to execute (such as run time support routines) are missing. Usually, partially resolved load modules are created by binding with the NCAL option. NCAL can save the cost of storing programs that are already stored in other libraries.

Binder control statements may aid program modification by replacing, deleting, and rearranging control sections. This means that an entire program need not be recompiled when there are changes to only a few of its control sections.

The Binder can also be used to construct overlay programs. This permits the programmer to decrease the amount of core a program requires by allowing those modules immediately needed to overlay modules not needed at that time.

The Binder also reserves storage for the common control sections generated by the Assembler and FORTRAN languages and the static external areas that are generated by PL/I.

For additional information on the Binder and its functions, consult the manual, *IBM DFSMS/MVS Program Management*, SC26-4916.

3.9.1 How to Estimate Region Size

In OS/390, storage is allocated separately for each job step. Therefore, each step can be restricted to the amount of core it needs.

The NIH Computer Center staff does careful research to select optimum region sizes for its cataloged procedures, and they should generally not be overridden without cause. Use of the maximum possible region may be unnecessarily expensive, especially for certain utility and service programs (like Sort/Merge and the Loader) which automatically format (use) all available region. No REGION parameter on a JOB or EXEC statement should request less than 16K.

Compile and bind steps for each language have what is generally sufficient storage assigned to them by the cataloged procedure. However, the storage required by the GO step is highly dependent on the program being executed. The default value of 4096K provided by the procedure may be too small, causing the job to ABEND. Because of this the user may want to determine the amount of storage a program needs before allowing it to run. This requires submitting a job to obtain a map of the fully resolved load module through use of the Binder. The job consists of a normal compile, followed by the appropriate LKGO step (e.g., FORVLKGO for VS FORTRAN), with COND.GO=(0,LE) on the EXEC statement.

Example 130

```
//stepname EXEC 111vCOMP
//COMP.SYSIN DD *
  (source program)
//stepname EXEC 111vLKGO,COND.GO=(0,LE)
```

where "111v" is the appropriate language and version prefix.

Include any Binder control statements needed for actual execution. Since the program will not be executed, there is no need to include any DD statements for the GO step.

The region size in bytes can be determined after this job is run by adding:

total length of program converted to decimal	(given in hexadecimal at the end of the module map.)
+n times buffer size of all data sets which can be opened at any one time	(buffer size = maximum block size.) The default for "n" is 5 for QSAM; 2 for all other access methods. Consult the applicable programmer's guide to determine which access methods are used by the job.
+20,000 bytes	(to cover supervisory services, control blocks, access method routines.)
+ enough to round this to the next higher even multiple of 1024 bytes.	

This figure, divided by 1024 (the value of K), can be entered on the execute statement invoking the LKGO procedure in the form:

```
REGION.GO=xxxxK   or   CORE=xxxxK
```

Example 131

```
//stepname EXEC FORVLKGO,CORE=2600K
```

For information on the maximum region size for each job class, see Section 2.2.3. When no region size is specified, the default is 4096K.

For more information on estimating region sizes, consult the applicable programmer's guide. PL/I users should also see the *PL/I Version 2 Programming Guide* for a discussion of the REPORT and ISASIZE execution time options.

3.9.2 Specifying Load Module Blocksize

Unless otherwise specified, the Binder will use for its output a blocksize that is equal to the output track capacity or 32760, whichever is smaller. This could be disastrous if the load module is to be copied to a device that has a smaller track size or to be exported to an installation, which does not have the device with the larger track size. The Binder will increase the BLKSIZE of an existing SYSLMOD PDS if the DCBS option is specified. For this reason, Computer Center procedures that add load modules to an existing PDS (e.g., CBLALKMM) use 13030 as the blocksize.

User jobs that invoke the Binder without executing a Computer Center procedure should specify the blocksize for the output load module rather than use the default. This can be done by coding the DCBS option of the Binder in addition to coding the blocksize for the load module in the DCB parameter of the SYSLMOD DD statement. When the DCBS option is

specified, the BLKSIZE value specified for the SYSLMOD data set will be used unless (1) it is larger than the maximum track size, in which case the maximum block size is used, or (2) it is less than the minimum blocksize (256), in which case the minimum block size is used. Any attempt to set the BLKSIZE of an existing SYSLMOD data set without specifying the Binder's DCBS option will be ignored.

Even when the DCBS option is specified, the Binder will not set the BLKSIZE for the SYSLMOD data set to a value less than the existing BLKSIZE.

Example 132

```
//stepname EXEC PGM=IEWL,PARM='MAP,LIST,DCBS'  
//SYSLIB DD ...  
//SYSLIN DD ...  
//SYSLMOD DD DSN=aaaaaaa.pdsname(pgm),DISP=(NEW,CATLG),  
//          DCB=BLKSIZE=13030
```

When preallocating a load module data set, always specify a BLKSIZE of 13030. PDSALLOC (see Section 3.6.5.1), TSO's ALLOCATE command, and WYLBUR's CREATE PDS command can be used for the preallocation.

3.10 USING THE LOADER

The Loader combines the bind and GO steps into one step. It will accept object modules from compilers or load modules produced by the Binder as input. It will also search libraries defined by the SYSLIB DD statement if unresolved external references remain after processing the primary input from SYSLIN. (To use the automatic search, programs must be stored under the same names by which they are called.) Binder control statements (INCLUDE, ENTRY, OVERLAY, etc.) cannot be used with the Loader.

Using the Loader is often more economical than using the Binder. The Loader does not create an intermediate data set; instead, the program is loaded directly into core and is executed from there. The Loader is best adapted to debugging runs where a different version of the program is compiled and executed each time. If the same program is to be executed many times, it is probably more economical to execute a fully-resolved load module (created with the llvLKSM or llvLKMM procedures) rather than using either the Binder or the Loader each time.

There is a Loader procedure for each language. The names of these procedures are of the form, llvLDGO. Examples using the Loader are provided in the individual NIH language manuals; *Using COBOL at the NIH Computer Center*, *Using FORTRAN at the NIH Computer Center*, *Using PL/I at NIH*, and *Using Assembler Language at the NIH Computer Center*.

The stepname for each procedure is GO, and all DD statements provided in the llvLKGO procedures are provided in the corresponding Loader procedures.

The entry point for the main program must be specified. Since the entry point is always PLISTART for PL/I, this name is in the PL/I procedures and does not have to be given. For FORTRAN the entry point is always MAIN unless the FORTRAN PROGRAM statement is used, in which case EPT=entry-point must be coded on the EXEC statement invoking the Loader procedure. For COBOL, the PROGRAM- ID is the default entry point unless a specific entry point is coded within the program. For Assembler, the entry point is coded in the program. To give the entry point, code EPT=entry point on the EXEC statement invoking the Loader procedure.

Example 133

```
//stepname EXEC CBLALDGO,EPT=REPORT7
```

The region size for each procedure is given in the language sections. The Loader runs in the same region as the program it processes and requires 22K. Therefore, the fully resolved program must be smaller than the region assigned to the LDGO step. Also, at least 2K (3K for FORTRAN, 8K for PL/I) of this region is required for tables and buffers. This number may become larger to accommodate extra table entries generated by the program being loaded. The remaining core is available to the problem program and any buffers it needs in execution. For example, a COBOL program which runs in 276K could, theoretically, be processed by the Loader in the default region of 300K (300K-24K=276K). Since the Loader uses the entire region assigned to it, using an unnecessarily large value may needlessly increase job costs. The Binder must be used if the programmer wants to know exactly how much core a program requires.

3.11 DSSUBMIT A PROGRAM TO SUBMIT BATCH JOBS

The DSSUBMIT program will read a data set from disk and submit the data set as a batch job. The format may be fixed or fixed block with an LRECL up to 254.

If a job should not be run unless a previous job terminates successfully, DSSUBMIT may be used as the last step of the first job to submit the second. Through condition code testing, the user can insure that the DSSUBMIT step does not execute if previous steps have failed.

Because using DSSUBMIT permits precise control of job executions for a series of jobs, it can be used to avoid data set contention problems (see Section 2.8).

Each data set submitted must contain all necessary JCL (including a /*KEYWORD statement). These statements must be standard JCL and cannot exceed 80 characters in length. However, the input stream data may be up to 254 characters long.

DSSUMIT stops processing as soon as JES2 informs it of an error; no additional data sets will be read. To insure that all jobs will be submitted, the JCL should include multiple DSSUBMIT steps, with each step submitting only 1 job.

The job control language to invoke DSSUBMIT to submit a cataloged data set is:

Example 134

```
//stepname EXEC PGM=DSSUBMIT
//SYSPRINT DD SYSOUT=A
//JOBxxxxx DD DSNNAME='aaaaiii.dsname',
//          DISP=SHR
```

Note that the ddname on the second DD statement must begin with the characters "JOB". Multiple "//JOBxxxxx" DD statements are permitted, and will cause multiple jobs to be submitted; each should have a unique DD name.

The data set read for input can also be a member of a cataloged PDS.

Example 135

```
//stepname EXEC PGM=DSSUBMIT
//SYSPRINT DD SYSOUT=A
//JOBxxxxx DD DSN=aaaaiii.dsname(member),
//          DISP=SHR
```

3.12 JCLSCAN TO CHECK FOR JCL ERRORS

Some of the most common reasons for job failure are JCL errors. JCLSCAN is a procedure that will cause the user's JCL to be scanned for syntactical errors without actually scheduling any part of the job; therefore no tapes or disks will be requested, and no programs will be executed. A JCLSCAN is usually an A class job.

Example 136

```
//stepname EXEC JCLSCAN
.
.
.
(user JCL)
.
.
```

NOTE: All JOB, JOBLIB DD, and /*MESSAGE statements must be removed before the job is submitted except for the Class A JOB statement needed for the run itself.

An in-stream procedure will not be scanned unless there is an EXEC statement for it.

The messages from this procedure explain themselves. The first diagnostic, IEF645I, is caused by a deliberate error (an invalid referback) in the procedure which forces the rest of the user's JCL to be scanned. All other diagnostic messages pertain to the user's JCL.

If the first diagnostic, IEF645I, does not appear in the output, the scan was incomplete because of one or more syntax errors. When these have been corrected, JCLSCAN should be run again to check for additional errors.

Using this procedure is more effective and less costly than testing JCL by executing a run. This is not a special program; it simply uses the standard OS Reader/Interpreter.

3.13 PASBILL FOR LISTING LAST MONTH'S COMPUTER BILL

The PASBILL procedure enables users to obtain a copy of the computer service bills for their accounts; the information is not broken down by initials. The cataloged procedure is used as follows:

Example 137

```
//stepname EXEC PASBILL
```

The program uses the account number on the JOB statement to produce the monthly bill for that account. If there are no charges for the given account during a month, the message:

```
"THERE WERE NO CHARGES FOR ACCOUNT aaaa DURING THE MONTH  
OF month"
```

will be printed. The data set is updated during the first 10 days of the month with the bills for the previous month.

As an alternative to the PASBILL procedure, the PAS ONLINE interactive accounting system, described in the *NIH Computer Center User's Guide*, displays summary reports via DB2. This charging information, available to users and account sponsors, is based on charge records for the full year.

Customers can also review their bills for CIT services on the NIH Data Warehouse via Data Town at:

<http://datatown.nih.gov>

Choose "DW Utilities" and then select "CIT Web Billing System."

3.14 GENERATION DATA SETS

CIT does not encourage the use of generation data sets, primarily because of the JCL and catalog difficulties in handling them. An excellent explanation of how to use generation data sets can be found in Appendix B of the manual *OS/390 MVS JCL User's Guide*, GC28-1758. For more advanced information, consult the *IBM DFSMS/MVS Using Data Sets*, SC26-4922 and *IBM FSMS/MVS Access Method Services for the Integrated Catalog Facility*, SC26-4906.

Briefly, a generation data set is one of a group of data sets cataloged chronologically. Numbers are used to indicate the relative "age" of a data set. The data sets may have like or unlike DCB attributes and data set organizations. If the DCBs and organizations of all

generations in a group are identical, they can be retrieved as one data set. Separate generations are referred to by coding the name of the generation group followed by a number in parentheses. A "+" in front of the number indicates a new generation added within the job. A "-" in front of the number indicates an older generation. If the number is a zero (with no sign), it indicates the current generation.

The generation numbers are updated at the end of the job. Therefore, generation data sets created and cataloged in a job are retrieved by their relative generation data set numbers throughout that job. (i.e., generation data sets +1 and +2 must be retrieved as +1 and +2 during that job.) If the job abnormally terminates and must be restarted, generation data sets that were created and cataloged before the restart step must have their relative generation data set numbers updated before the restarted job can be submitted. (i.e., generation data sets +1 and +2 created before the ABEND, must be referred to as -1 and 0 in the resubmitted job.)

Generation data sets may reside on disk or tape. All creation, deletion and modification of GDG members and their respective catalog entries must be performed by batch jobs. Any attempt to do otherwise, e.g., a WYLBUR or TSO command, can corrupt the structure and chronology of the entire GDG.

When a new generation is added to a generation data group (GDG), the data set name for the new generation must be entered in the catalog. This means that DISP=(NEW,CATLG) must be coded on the DD statement. If this is not done, duplicate data set names will be created in subsequent jobs.

The number of generations that are cataloged is determined by a number supplied when the generation data group index is created. When that number is reached, all the entries are uncataloged or the oldest entries only are either uncataloged or deleted. This option is one of those supplied when the generation index is created.

A model data set label or a data set used for this purpose must be created before a generation data group can be used.

3.14.1.1 The Model Data Set Label

A data set which is referenced by the operating system to obtain DCB attributes for new GDG data sets must exist before a new GDG data set is created. This data set, used as the model data set label, must be cataloged and must reside on a direct access volume. The data set name cannot contain special characters (except for periods used in a qualified name) and cannot be a generation data group name. The DCB parameter of the DD statement which describes the new GDG data set references the data set being used as the model.

If a label contains all the attributes required to define the new generation data set, only the name of the data set need be coded, (e.g., DCB=aaaaiiii.dsname).

If all the attributes required to define the new generation data set are not contained in the label, or if certain ones are to be overridden, follow the data set name with these attributes, (e.g., DCB=(aaaaiii.dsname,list-of-attributes)).

3.14.2 The Cataloged Index

After the model data set label has been created, a job must be run using the VDSUTIL procedure to build the index for the generation data group and define what action should be taken when the index overflows. The VDSUTIL procedure invokes Access Method Services (AMS); VDSUTIL and AMS are described in the manuals *Using VSAM and AMS at NIH*, *DFSMS/MVS Access Method Services (ICF)*, SC26-4906, and *DFSMS/MVS Using Data Sets*, SC26-4922.

The following example will create a generation data group index, with 3 entries, and delete anything older than 3 generations.

Example 138

```
//STEPX    EXEC   VDSUTIL
//SYSIN DD  *
DEFINE GDG (NAME(aaaaiii.NEWGDG.BASE) LIMIT(3) SCRATCH)
/*
```

SCRATCH should always be specified, otherwise the oldest version of the GDG will only be uncataloged, not scratched, and will continue to incur storage costs. LIMIT is a required parameter and must be specified.

The following example changes the number of generations for an existing GDG by setting a new LIMIT for the GDG.

Example 139

```
//STEPY    EXEC   VDSUTIL
//SYSIN DD  *
ALTER aaaaiii.NEWGDG.BASE LIMIT(6)
/*
```

3.14.3 Creating a New GDG Data Set

Once the data set used as the model data set label and the cataloged index have been established, the generation data sets may be created.

The following DD statement describes a generation data set named aaaaiii.UPDTE that will be written to a data set in the MSS management class. The data set aaaaiii.LABEL is used as the model.

Example 140

```
//ddname DD DSN=aaaaiii.UPDTE(+1),  
// DCB=aaaaiii.LABEL,UNIT=MSS,SPACE=(CYL,(5,2),RLSE),  
// DISP=(NEW,CATLG,DELETE)
```

The following DD statement describes a generation data set named aaaaiii.QTRLY which will be written to a 1600 BPI tape and sent to another installation. The data set aaaaiii.LABEL is used as the model. Since the model is a disk data set, the DEN subparameter of the DCB parameter must be added to the DD statement.

Example 141

```
//ddname DD DSN=aaaaiii.QTRLY(+1),  
// DCB=(aaaaiii.LABEL,DEN=3),UNIT=9TRACKHI,  
// DISP=(NEW,CATLG,DELETE),VOL=(PRIVATE,SER=tapeser)
```

3.14.4 Deleting a Generation Data Group

The procedure VDSUTIL is used to scratch and uncatalog generation data groups. The AMS DELETE command indicates the name of the generation data group.

In the following example, aaaaiii.NEWGDG.BASE is the name of the generation data group. This procedure will delete the GDG base index entry. All generations under this index entry must have been deleted before the index entry can be deleted.

Example 142

```
//STEPZ EXEC VDSUTIL  
//SYSIN DD *  
DELETE aaaaiii.NEWGDG.BASE GDG  
/*
```

3.15 EXECUTING WYLBUR COMMANDS AS A BATCH JOB

Note: A modified version of WYLBUR will eventually be placed on Titan. CIT encourages users to consider alternative long-term strategies through the WYLBUR Transition Working Group. Refer to *Interface* for information related to WYLBUR.

To keep informed about WYLBUR transition issues and to join the RETIRE-WYL listserv list go to:

<http://silk.nih.gov/silk/retirewyl>

WYLBUR commands can be executed as a batch job instead of online. This makes it possible to run lengthy command procedures, run command procedures that need to be executed at night, or deal with extremely large data sets. Executing WYLBUR commands as a batch job, rather than online, has several advantages:

- It frees up the terminal while WYLBUR commands for lengthy command procedures or large documents are executing.
- A large job can be run at a lower cost than if it were run interactively, especially if the batch job is submitted with the DISCOUNT option for off-hour processing.
- Up to 20 full-size active files can be used. Working interactively, the user is limited to just one full-size active file plus up to 19 smaller active files; the size of the remaining active files is more limited.
- WYLBUR can be used in conjunction with other programs in a series of operations. For example, you could execute another program to process some data, then invoke WYLBUR to send an electronic mail message notifying someone else that the job has been executed.

The input consists of the WYLBUR commands to be executed. To enter a BREAK, use the SET BREAK command (see the *WYLBUR General Editing* manual) to specify characters to be interpreted as a break signal. A break signal can only be entered as the first response to a prompt; it cannot be used to interrupt a listing, to abort the execution of a command or to ignore information typed.

Input to batch WYLBUR is from a DD statement with the name of SYSIN. Input has the following characteristics:

- It can be in either fixed (RECFM=F) or variable (RECFM=V) length records.
- There is no restriction to card format.
- It is not restricted to 80 columns.
- Input data sets cannot be in EDIT format.

The output looks like a complete WYLBUR session. The account and registered initials used for the "session" are those specified in the JOB statement.

There are two procedures: WYLBUR and WYLBURL. The only difference between the two is that WYLBURL permits the use of up to 50 active files with each active file using the maximum amount of space. Normally, the WYLBUR procedure should be used; use WYLBURL when several large active files are needed.

There is one optional symbolic JCL parameter available for the WYLBUR and WYLBURL procedures: OPTIONS, which indicates options to be used by the WYLBUR program. The following options are available:

OPTION	PURPOSE
ARGUMENT	<p>Passes a parameter to be made available to the WYLBUR commands being executed. The word ARGUMENT is followed by a string (a series of characters enclosed in quotes) containing the parameter. The parameter may also be enclosed in parentheses instead of quotes. The parameter is available as the value of the ARGUMENT function at command procedure level 0. (See the <i>WYLBUR Command Procedures</i> manual for more information on the ARGUMENT function and level 0).</p> <p>The ARGUMENT parameter can be made available to command procedures by passing it using the WITH option of EXECUTE, or by storing it in an active file or global variable.</p>
NO ABORT	Causes WYLBUR to continue execution of commands following an error. Normally, if an error occurs, the remaining WYLBUR commands in the input are not executed.
NO PROFILE	Specifies that the profile is not to be executed, as it normally would be. This is useful when WYLBUR is being invoked as part of a TSO CLIST or batch job used by many different people.
UNNUMBERED	Specifies that line numbers are not being used and that all positions of each line are to be treated as input to WYLBUR. Normally, line numbers are stored in the WYLBUR command lines when the job is submitted. In that case, if the input is either submitted as part of the RUN command or is read from an F format data set, the contents of lines in the last eight columns are ignored. If the input is a V or U format data set, the first eight columns are ignored.

OPTIONS are followed by an equal sign; the options are enclosed in single quotation marks and separated by commas.

Example 143

```
// EXEC WYLBUR,OPTIONS='NO ABORT',
// UNNUMBERED,ARGUMENT="ABC" '
```

The TO command works when running WYLBUR as a batch job. The message is sent to the specified user on both WYLBUR and TSO. For more information on the TO command, see either the *Fundamentals* or *General Editing* manual.

The SET TABS and SET TERMINAL commands are not used when running WYLBUR as a batch job. If either is used, an error message will be displayed.

In the following example, the large version of WYLBUR in the batch (WYLBURL) is used to obtain a formatted listing of a document. The document will be printed without line numbers on 999P (cut-sheet, 3-hole punch) paper. After the listing is submitted as a job, mail is sent indicating that the listing was submitted. The characters BRK are set as the break signal.

Example 144

```
// EXEC WYLBURL
SET BREAK BRK
LIST FROM MANUAL ON CAT MARKER # UNNUMBERED FORMS 999P
COLLECT
A paginated version of MANUAL has been listed.
BRK
MAIL TO III
```

3.16 HOST-TO-HOST FILE TRANSFER

On occasion there is a need to transfer files between the NIH Computer Center and other computer centers. This host-to-host file transfer is generally performed via two batch programs—SENDFILE and RCVFILE. Another file transfer program is CONNET:Direct, required by the Department of the Treasury for online financial transactions.

3.16.1 SENDFILE and RCVFILE

SENDFILE and RCVFILE can perform the file transfer under the following conditions:

- The other site (computer center) must be defined to NIH as a JES2 NJE node; i.e., it must be an OS/390 JES2 or JES3 node, a VM RSCS node, or a VSE POWER node.
- The other site must have the NIH Computer Center-written SENDFILE and RCVFILE programs (or equivalent programs for VM or VSE).
- Users transferring a file must be authorized to use both the sending and receiving site.
- Any job that is submitted from the South System to run on Titan must have
",USER=userid " coded on the JOB statement (where "userid" is the Titan RACFid under which the job will run) and
_PASSWORD=password coded on the JOB statement (where password is the Titan password for the RACFid under which the job will run)
- In order to view the job in OUTPUT HOLD on Titan instead of at the South System, add a / *ROUTE PRINT NIHJES2 statement.
- Users who run jobs at other NJE nodes and then route the job to print on Titan must add a / *JOBPARM ROOM=bbbb statement after their JOB statement where "bbbb" is the box number where the output is to be placed. If you want this output placed in one of the boxes at the Parklawn site, place the letter "P" in front of the box number.

Currently the following nodes are defined to the South System:

FSAJES2	ACF, Baltimore, MD
CCF1	BLS, SunGard, Voorhees, NJ
NPH	BLS, SunGard, Voorhees, NJ
CDCJES2	CDC, Atlanta, GA
SPGFLD	DOC, Springfield, VA
TCCB	DOT, Washington DC
HCFJES	CMS, Baltimore, MD
NIHJES2	NIH Titan
SSAPRD1	SSA, Baltimore, MD

The above nodes have the SENDFILE and RCVFILE programs; other defined nodes may obtain them upon written request from the NIH Computer Center (non-OS/390 nodes will have to modify them to suit their environment).

The SENDFILE and RCVFILE programs are used, in conjunction with the /*XMIT JCL statement (described in Section 2.10.8), to perform host-to-host file transfer. SENDFILE reads the data set to be transferred and converts it into 80-byte records for transmission, adding the necessary control information to permit the data set to be reconstructed at the receiving site. It also accepts a job stream to be executed at the receiving site (to reconstruct the data set). RCVFILE reconstructs a data set from the 80-byte records created by SENDFILE. The RCVFILE program is always executed at the receiving site (host).

Both the SENDFILE and RCVFILE programs require SYSUT1 and SYSUT2 DD statements to define input and output. In addition, SENDFILE requires a special JCL DD statement to define the job stream that will be executed at the receiving site (host) to reconstruct the file transmitted by SENDFILE.

The following examples and commentary illustrate host-to-host transfer of a simple sequential data set, and host-to-host transfer of an entire PDS. Note that it is possible to transfer nonsequential data sets by first creating a sequential, transportable copy (e.g., an IEBCOPY unload of a PDS or an AMS EXPORT or REPRO of a VSAM data set).

In both examples below, the sending host (site) is the NIH Computer Center; the receiving host (site) is another OS/390 node identified as nodename. The JCL starting with the JOB statement down to and including the /*XMIT statement must conform to the operating system and standards of the sending site (in these examples, NIH). The JCL starting with the "rcvjob" JOB statement must conform to the operating system and standards of the receiving site (nodename). (NOTE—The DLM=XX and XX statements must be written exactly as shown [XX could be any unique character sequence]). The following example can be used for the transfer of a sequential data set.

Example 145

```
//IIISEND1 JOB ...
//SEND EXEC PGM=SENDFILE
//SYSUT1 DD DSN=AAAAIII.DATASET,DISP=SHR
//SYSUT2 DD SYSOUT=(A,INTRDR)
//JCL DD DATA,DLM=XX
//IIIXMIT JOB (NIH JOB statement)
/*KEYWORD=kkk
/*XMIT nodename
//rcvjob JOB (receiving node JOB statement)
(any other control statements required by receiving host)
//RECEIVE EXEC PGM=RCVFILE
//SYSPRINT DD SYSOUT=*
//SYSUT2 DD (parameters defining reconstructed data
set)
//SYSUT1 DD *
XX
```

In the SEND jobstep, the SYSUT1 DD statement defines the sequential data set to be transferred. The SYSUT2 DD statement defines the output containing the job stream and converted data set and is normally coded as shown. The JCL DD statement defines a job stream to be executed at the receiving host to reconstruct data set AAAAIII.DATASET at receiving site nodename. (NOTE: This jobstream must always include a job step that executes the RCVFILE program.)

In the RECEIVE jobstep, the SYSPRINT DD statement is required and defines a message data set. The SYSUT2 DD statement is required for RCVFILE and defines the data set that was transmitted and is to be reconstructed at the receiving site (host). The SYSUT1 DD statement is required and must be the last JCL statement prior to the delimiter XX.

The following example can be used for the transfer of an entire PDS.

Example 146

```
//IIISEND2 JOB (NIH JOB statement)
//UNLOAD EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//IN DD DSN=AAAAIII.PDS,DISP=SHR
//OUT DD DSN=&&TEMP,DISP=(,PASS),UNIT=SYSDA,
// SPACE=(CYL,(1,1),RLSE)
//SYSIN DD *
COPY INDD=IN,OUTDD=OUT
/*
//SEND EXEC PGM=SENDFILE,COND=(0,NE)
//SYSUT1 DD DSN=&&TEMP,DISP=(OLD,DELETE)
//SYSUT2 DD SYSOUT=(A,INTRDR)
//JCL DD DATA,DLM=XX
//IIIXMIT JOB (NIH JOB statement)
/*KEYWORD=kkk
/*XMIT nodename DLM=YY
```



```

//rcvjob    JOB    (receiving node JOB statement)
              (any other control statements required by receiving host)
//RECEIVE   EXEC   PGM=RCVFILE
//SYSPRINT DD    SYSOUT=*
//SYSUT2    DD     DSN=AA&TEMP,DISP=(,PASS),UNIT=SYSDA,
//              SPACE=(CYL,(1,1),RLSE)
//SYSUT1    DD     *
**SENDFILE**
/*
//LOAD      EXEC   PGM=IEBCOPY,COND=(0,NE)
//SYSPRINT DD    SYSOUT=*
//IN        DD     DSN=AA&TEMP,DISP=(OLD,DELETE)
//OUT       DD     (parameters defining reconstructed data
set)
//SYSIN     DD     *
              COPY INDD=IN,OUTDD=OUT
/*
YY
XX

```

The first step of this job unloads the PDS named AAAAIII.PDS to a temporary sequential file named AA&TEMP, which is then passed to the next job step (named SEND) for transmission to the receiving host named nodename. The SENDFILE step will not be executed if something goes wrong with IEBCOPY (controlled by COND=(0,NE) on the EXEC statement).

(NOTE: As contrasted with the first example in this section, the /*XMIT statement must contain a delimiter—DLM=YY—because the data stream that follows /*XMIT itself contains statements beginning with /*.)

In this example, the special control statement **SENDFILE** must be included immediately after the SYSUT1 DD statement, and must be coded as shown in columns 1/12. The **SENDFILE** record signals SENDFILE to insert the transferred data set (AA&TEMP) at this point. This is necessary because another job step follows RCVFILE. The last job step executes IEBCOPY to reconstruct the transmitted sequential data set named AA&TEMP into a PDS.

SENDFILE DD Statements

SYSUT1	Required. Variable-length spanned records cannot exceed 32760 bytes.
SYSUT2	Required. RECFM=FB,LRECL=80,BLKSIZE=80 (default) or multiple of 80. Normally defined as SYSOUT=(A,INTRDR).
JCL	Required.

RCVFILE DD Statements

SYSPRINT	Required. RECFM=FBA,LRECL=121,BLKSIZE=121 (default) or multiple of 121.
SYSUT2	<p>Required. The RECFM, LRECL, and, if not overridden, BLKSIZE will be those of the original data set. If a new BLKSIZE is specified, the following should be noted:</p> <ul style="list-style-type: none">• For fixed-length records, if BLKSIZE is not equal to LRECL, the "blocked" indicator will be set.• For variable-length records, if BLKSIZE is less than LRECL+4, the "spanned" indicator will be set; otherwise, the "spanned" indicator will be turned off.• For undefined-length records, the new BLKSIZE must be at least as large as the original BLKSIZE; otherwise, a S013-34 ABEND will occur.

SENDFILE and RCVFILE Completion Codes

SENDFILE returns a completion code of 0 if SYSUT1 does not define a null data set, and 4 if it does. SENDFILE issues a U0100 ABEND if a DD statement is omitted.

RCVFILE returns a completion code of 0 if the data set is successfully reconstructed, 4 if SYSUT1 defines a null data set, and 12 if an error occurs (forcing RCVFILE to terminate). The SYSPRINT message indicates the execution result. RCVFILE issues a U0100 ABEND if a DD statement is omitted.

3.16.2 CONNECT:Direct

CONNECT:Direct is a product that provides host-to-host file transfer. It is required by the Treasury Department for online financial transactions with their systems. The function it provides is similar to that provided by the SENDFILE and RCVFILE programs (see Section 3.16.1). CONNECT:Direct is easier to use than those two programs and provides a means to monitor the progress of the file transfer.

There are two requirements for use of CONNECT:Direct: there must be a VTAM or TCP/IP connection between NIH and the other site, and the other site must also have the hub version of CONNECT:Direct.

The actions carried out by CONNECT:Direct are controlled by a CONNECT:Direct process, which is similar to a JCL procedure. Once the process is set up, the user needs to supply just the values for various parameters. The two parameters used most frequently are the source data set name and the target data set name.

The NIH Computer Center's CONNECT:Direct node names, APPLids, and NETids are:

System	Node Name	APPLid	NETid
South System	nih.ndm	NIHNDM	NIH
Titan	ndm.std.nih	NIH4ND	NIH

Anyone wishing to set up a new CONNECT:Direct application must be a registered CIT user. Contact TASC or the account sponsor for your organization if you do not already have a CIT account/initials combination (CIT USERid). Once you are a registered CIT user, you can then register for using CONNECT:Direct. Use of CONNECT:Direct requires coordination with another site as well as modifications to certain CONNECT:Direct configuration files.

To register to use CONNECT:Direct, submit an SRT that includes the following information:

- destination of the data to be transmitted
- nature of the data to be transmitted
- the amount and frequency of transmitted data

Each project must have a Data Transmission Administrator (DTA), who is responsible for setting up and maintaining the jobs and CONNECT:Direct processes needed to transmit the data. All correspondence with CIT regarding a registered CONNECT:Direct project should specify the name of the DTA.

Data Transmission Administrators should refer to the following manuals:

CONNECT:Direct for MVS User's Guide

This manual introduces the command system through which the CONNECT:Direct Processes are controlled. It describes the commands available and tells how to use them through the Batch Interface. The manual assumes knowledge of the IBM Multiple Virtual Service operating system and job control language.

CONNECT:Direct Process Guide

This manual provides information needed to write CONNECT:Direct Processes. A CONNECT:Direct Process is a unit of work. Processes are written in a language unique to the CONNECT:Direct product. The CONNECT:Direct Process language consists of statements and parameters that provide instructions for initiating such activities as copying files from one CONNECT:Direct node to another.

The two manuals listed above can be obtained from the vendor, Sterling Commerce (<http://www.sterlingcommerce.com/>).

Only the base CONNECT:Direct product is installed on the South system of the NIH Computer Center. South system access to it is available only through the batch interface.

Following is the JCL to be used for submitting a CONNECT:Direct job on the South system. The `/*ROUTE XEQ VTAM1` statement is required so that the CONNECT:Direct job will run on the correct processor. In the following example, `aaaa` and `iii` refer to the user's own account number and registered initials:

```
(job card)
(*ROUTE XEQ and /*MESSAGE statements for any tape
input data sets)
/*ROUTE XEQ VTAM1
//NDM EXEC PGM=DMBATCH,REGION=1024K,PARM=(YYSLYNN)
//STEPLIB DD DSN=ZZXZNCP.NDM.LINKLIB,DISP=SHR
/* DDNAMES BELOW MUST BE SPECIFIED EXACTLY AS SHOWN
//DMNETMAP DD DSN=ZZXZNCP.NDM.NETMAP,DISP=SHR
/* LIBRARY CONTAINING CONNECT:DIRECT PROCESSES
//DMPUBLIB DD DSN=AAAAIII.libname,DISP=SHR
//DMMSGFIL DD DSN=ZZXZNCP.NDM.MSG,DISP=SHR
//DMPRINT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
```

Restrictions

Because a data transmission job initiated at another site would have no way of specifying a job class or supplying a `/*ROUTE XEQ` or `/*MESSAGE` statement, it is not possible to initiate a data transmission from another site to a tape data set at NIH. It would be possible, though, to initiate a job at NIH, which transmits data from another site to here. In that situation, because the job initiated at NIH could specify an appropriate job class and `/*ROUTE XEQ` and `/*MESSAGE` statements, data from another site could be placed into a tape data set at NIH.

* CONNECT:Direct commands go here. This is a CONNECT:Direct comment.

4 SOUTH – MESSAGES AND CODES

This section includes explanations of many of the messages users can receive. The last subsection, 4.6, has a list of publications containing documentation of standard IBM messages.

4.1 ABENDS AND CRITICAL MESSAGES

This section includes explanations of ABENDs and critical messages.

4.1.1 Common ABENDs

When computing tasks end abnormally (called an ABnormal END, or ABEND), the system provides ABEND messages that describe the problem, with "return codes" that pinpoint the problem more specifically. Here is a list of some common ABEND codes along with brief descriptions of their meanings for typical return codes.

- | | |
|-----|--|
| 322 | the batch job exceeded the amount of CPU time requested in the JOB statement or, if no CPU time request was specifically coded in the JOB statement, exceeded the default amount of CPU for the JOB class. Jobs that get this ABEND either need more CPU time to complete the computing tasks or contain an error that caused the job to loop endlessly through a set of instructions until the CPU time maximum was reached. |
| 222 | the batch job or TSO session was cancelled. The cancel request can come from the user (via WYLBUR's CANCEL command) or from one of the computer operators at the NIH Computer Center complex. A message appears with the ABEND describing why the job was cancelled, such as "JOB CANCELLED AT USER REQUEST." |
| 722 | the batch job tried to produce more lines of output than requested in the JOB statement or, if no output lines request was specifically coded in the JOB statement, exceeded the default amount of output lines for the JOB. The solution for this ABEND is to ensure that you want to print all the output in question (the default maximum is 20,000 lines) and, if so, to increase the output lines request in the JOB statement. |
| F82 | the batch job had a problem when attempting to do tape processing. The message produced with this ABEND (such as "NO /*MESSAGE FOR TAPE" or "NL SCRATCH DISALLOWED") describes the problem and indicates the area in the job that must be changed for successful completion. |
| 80A | the batch job or interactive program ran out of memory (REGION). Usually this can be remedied by increasing the REGION request for the job or job step that ABENDED. Section 3.9.1 contains information on how to determine the approximate amount of region needed. |

-
- 213 unable to locate a data set (and thus was also unable to open it). Perhaps the name was specified incorrectly or the wrong data storage volume was indicated in the JCL of the job. The solution is to find (or create) the data set needed by the job and to make sure the JCL points to the data set correctly.
- 613 unable to open a tape data set due to an I/O (input/output) error on the tape. The data set could not be read from or written to the tape as requested by the job. The solution is to find out what is on the tape (via TAPEMAP) and ensure that the job isn't trying to do something impossible (like write a 5th data set on a tape that doesn't contain a 4th data set). If a flaw in the magnetic recording media of the tape is causing this problem, assistance might be required. Contact the CIT Technical Assistance and Support Center (TASC) if there is a problem.
- 813 tried to open a tape data set, but the data set name indicated in the JCL doesn't match the data set name on the tape. Solution: find out the real name of the tape data set (via TAPEMAP) and correct the JCL.
- 0C1 the program received an invalid operation code, usually the result of a "wild branch" in the program or a missing DD statement in the JCL.
- 0C4 the program tried to write in an area of memory that doesn't belong to the job, also called a "protection exception." Frequently this is the result of the job trying to write to an array location that is greater than the size defined for the array. The solution is to check the program to find out where the array location is going out of bounds.
- 0C7 the program (usually in COBOL) attempted to put "bad data" in a packed decimal field, resulting in this "data exception" ABEND. The solution is to ensure that the data is in the format that the program was written to handle.
- B37 the batch job or interactive program couldn't complete output to a data set because the amount of disk space allocated for the data set wasn't sufficient. The solution is to make sure that the output disk data set is given a large enough primary allocation. It is advisable to make appropriate allocations in the primary space allocation.
- x13 (where "x" is a number or a letter up to F) the batch job or interactive program could not open a data set needed by the job.
- 413-1C the TAPESCAN utility was run against a second or subsequent volume of a multi-volume data set. TAPESCAN cannot be used for tapes that include multi-volume data sets.

The meaning of ABEND messages and return codes can be found in the manuals *OS/390 MVS System Messages, Volume 1* (GC28-1784), *Volume 2* (GC28-1785), *Volume 3* (GC28-1786), *Volume 4* (GC28-1787), and *Volume 5* (GC28-1788), available through the CIT Technical Information Office. Refer to Section 1 of this manual for information on ordering publications. If you have questions about an ABEND received by a batch job or need assistance correcting an ABENDING batch job or interactive program, contact TASC.

4.1.2 Critical Messages

IEA794I SVC DUMP HAS CAPTURED, SUBMIT SRT ASAP

- Cause: The job has caused a serious system problem.
- System Action: The job failed. The output will be HELD FOR SYSTEM DOCumentation and not purged from the system until CIT releases it.
- Correction: Contact TASC immediately for help or submit a "Critical" Service Request Ticket. Save all job output and forward it to CIT staff. Freeze all data sets. Do not re-run the job until the staff informs you that the problem has been resolved.

CATASTROPHIC DATA FILE ERROR

- Cause: From a terminal the user issued a WYLBUR command that encountered a bug and caused a system problem.
- System Action: The user's WYLBUR session is suspended until it can be recovered.
- Correction: When the session is recovered the user must **not** reissue any of the last few commands. Contact TASC immediately. Print your screen or write down the last few commands you typed. Forward all documentation to the TASC staff. Freeze all data sets. Do not reissue the commands until you are informed by TASC that the problem has been resolved.

operator message . . .CONTACT TASC 301-594-6248

- Cause: This message is sent whenever an operator cancels your job; it explains why the job was cancelled.
- System Action: The job was flushed.
- Correction: Contact TASC only if you do not understand the operator's message and do not know what caused the problem; otherwise, correct the job and resubmit.

4.2 GENERAL MESSAGES (NIH, HASP, IEF, MIM)

MIM1038 jobname2 CONTENTION WITH jobname1 OWNS type ON cpuid

Cause: Your job (jobname2) required access to a data set that was already controlled (owned) by another session or job (jobname1).

The data set name is listed in the following MIM1039 message. This message is issued when a job enters the system and attempts to enqueue on a data set name which is already enqueued upon in any of the following ways:

Owner's disposition

NEW, OLD, or MOD
SHR

Your job's disposition

OLD, SHR, NEW, or MOD
OLD, NEW, or MOD

System Action: Your job was suspended until the data set became available.

Correction: Avoid data set contention by:

- Use DISP=SHR whenever the data set is to be read and not modified.
- Use unique data set names even for work files and for data sets on different volumes.
- When more than one job may access the same data set, use either /*BEFORE and /*AFTER or /*CNTL statements to reduce contention problems. See Section 2.8.

MIM1039 jobname2 NEEDS type name dsname

Cause: This message is listed following the previous one and is repeated for every data set in this job that is in conflict with other jobs or terminal sessions.

See the previous message for "System Action" and the appropriate "Correction."

NIH921E - JOB CANCELLED BEFORE EXECUTION; MIM MESSAGES SHOW
DATA SET CONTENTION

Cause: The job was suspended for data set contention for more than 30 minutes.

System Action: The job was cancelled before executing any programs or accessing any files.

See the MIM messages described above for further information and appropriate "Correction."

NIH984E - RACF PROTECTED DATA SETS CANNOT BE RECOVERED

Cause: The user ran an ADSRECOV job or used WYLBUR's ENTER RECOVER command to recover a RACF-protected data set from a backup of a dedicated disk volume.

System Action: The job fails with a return code of 12.

Correction: RACF-protected data sets cannot be recovered from backups of dedicated disk volumes.

NIHUTL01 - jobname HAS BEEN IN A WAIT STATE SINCE hh:mm:ss

Cause: The job was inactive for a period of time. The time given, in hours, minutes, and seconds, is the approximate time the job became inactive. Possible causes of the delay include waiting for a response from the operator, hardware malfunction, or programming error.

System Action: If the problem is resolved, the job resumes execution. If the problem cannot be resolved, the job may be cancelled.

Correction: Correction is needed only if the job did not run to completion.

\$HASP185 OUTPUT INCOMPLETE

Cause: The system could not properly close a SYSOUT data set before job termination. Some reasons for this are:

- The job was in the system when the system crashed and RERUN was not requested by the user. In this case, the message "JOB WAS EXECUTING WHEN SYSTEM FAILED - OUTPUT DATA SETS MAY BE INCOMPLETE" may appear after the JES2 JOB LOG, the message "JOB DELETED BY JES2 OR CANCELLED BY OPERATOR BEFORE EXECUTION" may appear in the JES2 JOB LOG, or both may appear. The type of system failure dictates which message(s) will be issued.
- Spooling I/O errors occurred during execution or printing, and the output could not be completed.

A system error occurred, and the SYSOUT data set was not closed properly.

\$HASP170 device name DELETED

- Cause: The user, at a terminal or an RJE workstation, or a computer room operator requested that the job be purged while in the process of printing.
- System Action: The job will be terminated shortly after the command is issued, and the message will appear at the end of the SYSOUT output.
- Correction: None is needed; the message is for informational purposes only.

\$HASP170 device name action

The following messages may appear at the end of a page of SYSOUT output. They are generated as a result of action taken by an operator.

- BACKSPACED the operator has requested that a portion of the output be reprinted.
- FORWARD
SPACED the operator has requested that a portion of the output be skipped.
- RESTARTED the operator has requested that the output for this job be terminated and reproduced from the beginning.
- REPEATED the operator has requested that an additional copy of the output for this job be produced.
- INTERRUPTED the operator has requested that the output be suspended and restarted later from the point at which it was interrupted.
- TRANSLATED the operator has requested that nonprintable characters be translated to blanks before a line is printed. This will not affect the appearance of the output, but may allow the printer to go faster.
- SPACED the operator has requested that carriage control be ignored, and that all lines be single spaced.

\$HASP377 jobname FORCIBLY TERMINATED FOR EXCEEDING LINES BY
80000

- Cause: A dump was generated for debugging purposes and a loop occurred in ABEND processing.
- System Action: The system will terminate any job which exceeds its line estimate by 80,000 lines.

Correction: None is needed; the message is for informational purposes only.

4.3 DISK AND MSS MESSAGES

NIH300I JOB CANCELLED--EXCESSIVE ALLOCATION OF SCRATCH DISK SPACE

Cause: The job allocated more than 100,000 tracks of SYSDA.

System Action: The job was cancelled.

Correction: Reduce the scratch space request and resubmit.

NIH300I WARNING: THIS JOB APPROACHES THE SCRATCH DISK SPACE ALLOCATION LIMIT

NIH300I ANY JOB THAT ALLOCATES OVER 100,000 TRACKS OF SCRATCH DISK SPACE WILL BE CANCELLED

Cause: The job allocated more than 60,000 tracks of SYSDA.

System Action: The job continues.

Correction: No action is required; but user should ensure the job does not exceed the 100,000 track SYSDA maximum.

NIH802E NONSTANDARD DATA SET NAME - ILLEGAL: dsname

Cause: The job has attempted to allocate a data set with a nonstandard data set name.

System Action: The job was terminated.

Correction: The data set must be renamed to agree with standard IBM conventions; see Section 2.5.8 for details.

NIH801W WARNING -- NONSTANDARD DATA SET NAME: dsname

Cause: The job has attempted to open a data set with a nonstandard data set name.

System Action: The job continued.

Correction: The data set must be renamed to agree with standard IBM conventions; see Section 2.5.8 for details.

NIH813I JOB FAILED DUE TO ILLEGAL USE OF AMS COMMAND - cmd

where "cmd" is the AMS command causing the ABEND.

Cause: (1) If cmd is DEFINE, then the problem could be that an illegal data set name was specified for the data or index component when defining a cluster or alternate index.
(2) The command specified or one or more of its operands are not supported.

System Action: The job will fail with an ABEND S047.

Correction: (1) Correct the data set name and resubmit. See Section 2.5.8 for VSAM naming conventions.
(2) Correct the job and resubmit. See the *Using VSAM and AMS at NIH* preface for AMS commands and operands supported.

NIHDS06 INVALID DSORG - [UNMOVABLE OR ISAM] DS NOT ALLOWED ON THIS DISK

NIHDS06 DSN=xxxxxxx

Cause: This type of data set is not allowed on the requested class of volumes.

System Action: The job was cancelled.

Correction: Specify an appropriate volume and resubmit.

NIHDS07 INVALID SPACE REQUEST - [ABSTR OR MXIG OR ALX]

NIHDS07 DSN=aaaaiii.name

Cause: This type of allocation is not allowed on public volumes.

System Action: The job terminated.

Correction: Change the space request to omit the invalid specification.

NIHDS09 NOT MORE THAN nnn CYLINDERS ALLOWED ON type VOLUMES

NIHDS09 DSN=aaaaii.name

Cause: For the specified data set the job requested more than the maximum space allowed on the given type of volume.

System Action: The job terminated.

Correction: Change the SPACE parameters so that the space request is valid for the type of volume.

NIHDS13 DISPOSITION OF CATALOG REQUIRED FOR MSS

Cause: Normal or conditional disposition of a new MSS data set was incorrectly specified.

System Action: The job failed.

Correction: Correct so that the normal termination disposition is CATLG and the conditional disposition is CATLG (the default) or DELETE.

NIHDS27 MULTI-VOLUME DATASETS NOT ALLOWED ON PUBLIC VOLUMES

NIHDS27 DSN=aaaaiii.NAME

Cause: The job attempted to allocate a multi-volume data set on public direct access volumes.

System Action: The job terminated.

Correction: Change the request to one volume.

NIHDS30 - dsn IS NOT A VALID DATA SET NAME FOR PUBLIC VOLUMES

Cause: The job tried to allocate a data set with an invalid name on a public volume or attempted to place a temporary data set on a public volume.

System Action: The job terminated.

Correction: Specify an appropriate data set name.

The following IBM messages for the MSS are included here because documentation is not complete in any publication being made available to users.

IEF287I dsname-NOT CATLG 2 **DATA SET WILL BE SCRATCHED IF ON
MSS

- Cause: The job was unable to catalog a data set because the name was invalid for the catalog or already existed.
- System Action: The job fails with a JCL error. The data set was created and kept but not cataloged. If an MSS data set, it will be scratched by CIT but not until regular MSS maintenance.
- Correction: The DSRENAME procedure should be used to rename and catalog the data set or the DSSCR procedure should be used to scratch the data set.

IEF643I UNIDENTIFIED POSITIONAL PARAMETER IN THE DISP FIELD

- Cause: An improper conditional disposition was coded for a new MSS data set.
- System Action: The job failed.
- Correction: Code a normal disposition of CATLG (e.g., DISP=(NEW,CATLG)).

IGD31GI DATA SET ALLOCATION REQUEST FAILED - DATA SET -
' dsname ' NAMED IN QUOTES, IS NOT ELIGIBLE TO BE SMS-
MANAGED

- Cause: The job specified a name for a new data set on a public volume, using single quotes.
- System Action: The job failed.
- Correction: Remove the quotes surrounding the data set name.

4.4 TAPE HANDLING MESSAGES

NIH130I F82-04: serial, NO /*MESSAGE FOR TAPE

- Cause: Your job requested access to the tape whose serial number is given in the message but did not include a /*MESSAGE statement for that tape.

System Action: The job automatically terminates (ABENDs) with a system code F82.

Correction: Resubmit the job including the /*MESSAGE statement.

NIH130I F82-08: serial, WRITE DISALLOWED ON READ-ONLY TAPE

Cause: Your job tried to write on a tape which had its status set to READ only.

System Action: The job ABENDs with a system code F82.

Correction: Change the status of the tape to WRITE before resubmitting the job. (The VOLSTAT procedure is described in Section 3.5.6.)

NIH130I F82-12: serial NL SCRATCH DISALLOWED

Cause: The job requested that an unlabeled scratch tape be mounted. Unlabeled scratch tapes are not provided.

System Action: The job ABENDs with a system code F82.

Correction: Change the JCL to use labeled scratch output tapes.

NIH130I F82-16: serial "W" NOT SPECIFIED FOR OUTPUT TAPE

Cause: Your job OPENed a tape for output, but the /*MESSAGE statement for that tape did not include a "W" to permit writing.

System Action: The job ABENDs with a system code F82.

Correction: Since "R" for read only is the default, "W" must be included on the /*MESSAGE statement for each tape which will be OPENed for output.

NIH130I F82-52: serial UNPROCESSED MICRO TAPE

Cause: The tape has been written and is waiting for microfiche processing.

System Action: The job ABENDs with a system code F82.

Correction: Resubmit the job after the MICRO output has been created or cancelled.

NIH130I F82-60: TAPE MOUNTS ARE ILLEGAL IN THIS JOB CLASS

- Cause: Your Class A or E job tried to use a tape.
- System Action: The job ABENDs with a system code F82.
- Correction: Change the JOB statement to request a job class which does permit the use of tapes (e.g., Class B), or drop the requirement for a tape.

NIH130I WARNING: serial "W" NOT SPECIFIED FOR INOUT TAPE

- Cause: The indicated tape was opened for input and output even though the intention was only to read it. This most commonly occurs during the execution of a FORTRAN program when the user neglected to code IN as part of the LABEL parameter on the DD statement.
- System Action: The warning message is issued and the job continues.
- Correction: Include LABEL=(n,SL,,IN) or LABEL=(,,IN) on the DD statement.

NIH130I WARNING: serial ASSIGNED TAPE, MOUNT SCRATCH

NIH130I WARNING: serial MICRO TAPE, MOUNT SCRATCH

NIH130I WARNING: serial UNKNOWN, MOUNT SCRATCH

- Cause: These three warning messages are the result of operator errors in mounting an assigned, MICRO, or "special" (UNKNOWN) tape when a scratch tape should have been mounted.
- System Action: The operator corrected the error by mounting a scratch tape and the job continued.
- Correction: No action is required by the user.

CTT143W: jobname, CONTROL-T INFORMATIONAL WARNING MESSAGE
(rc)

CTTX005: ddname DSNAME=dsname
ALREADY EXISTS OR MAY CAUSE OVERWRITING
OF PREVIOUS DATA ON TAPE nnnnnn. SEE
&PUBLIC.TAPES FOR DETAILS.

- Cause: These are system warning messages from the tape automation system (e.g., if a user writes over an existing data set).
- System Action: The warning message is issued and the job continues.

Correction: No immediate action is required by the user. See &PUBLIC.TAPES.

ONLY UNIT/DRIVE DESIGNATIONS OF 9TRACKHI OR CTAPE VALID WITH
SPECIAL TAPES

CORRECT UNIT/DRIVE FIELD FOR SPECIAL TAPE *** serial *** AND
RESUBMIT

Cause: A unit of TAPE has been incorrectly used for a tape.

System Action: The job is cancelled before execution.

Correction: Resubmit the job with the correct volume serial number or add the
correct UNIT parameter in your JCL to specify the appropriate type
of tape.

TAPE serial SPECIFIED ON /*MESSAGE STMT IS NOT AN ASSIGNED
TAPE

Cause: The tape whose serial number appears on the /*MESSAGE statement
is unassigned.

System Action: The job is cancelled before execution.

Correction: Resubmit the job with the correct volume serial number.

TAPE serial SPECIFIED ON THE /*MESSAGE STMT IS NOT IN THE TAPE
LIBRARY

Cause: The tape library inventory system shows that the tape is not in the
tape library. It may have been checked out of the library, it may be a
"special" non-library tape, or the number may have been given
incorrectly.

System Action: The job is cancelled before execution.

Correction: Specify the tape request correctly, and resubmit the job.

CTT108E 0545, PREVIOUS FILE DOES NOT EXIST ON VOLUME nnnnnn

Cause: This data set has not yet been recorded in the tape management
system database. Whenever a data set is read or a new data set is
written, its name and sequence number are stored in the database.

System Action: The job ABENDs with a system code U242.

Correction: Run the TAPESCAN procedure to open all the data sets on a single tape so information can be added to the tape management system database.

```
NIHCTT1: ****Warning**** VOLSER = serial OPEN FOR OUTPUT
NIHCTT2: Tape Dataset name does not start with a valid AAAAIII
NIHCTT3: JOBNAME = jobname DDNAME = ddname
NIHCTT4: DSN=dsname
NIHCTT5: For more information see URL silk.nih.gov/tapedsn
```

Cause: These warning messages are a transitional aid to prepare for the migration to Titan. The system issues these messages when reading or writing a tape data set (DSNAME) that does not begin with a valid account/initials combination. On Titan, all data sets must begin with a Titan USERid.

System Action: These five warning messages are issued and the job will continue.

Correction: Specify a tape data set name that begins with a valid account/initial combination.

4.5 JOB STATEMENT ERROR MESSAGES

The following messages indicate that an error in the JOB statement caused the job to be cancelled immediately when it was submitted. The job may be submitted again as soon as the error is found and corrected.

**** ILLEGAL JOB STATEMENT ****

**** JOB NAME MISSING ****

**** JOB NAME TOO LONG ****

**** NO PARAMETERS FOR JOB ****

**** PARMS MUST BE IN PARENS ****

**** PROGRAMMER NAME MISSING ****

**** INVALID CLASS SPECIFICATION ****

**** INVALID PARAMETER ON JOB STATEMENT ****

**** ILLEGAL CONTINUATION ****

**** INVALID JOB CLASS ****

**** MISSING ACCOUNT NUMBER ****

**** MISSING BOX NUMBER ****

**** ERROR IN ACCOUNTING FIELDS ****

**** INVALID CHAR IN BOX NUMBER ****

4.6 WHERE TO FIND OTHER MESSAGES

The following table shows the title and form number of the publication that describes each type of message.

Figure 4-1. Publications Containing Messages

Message	Publication	Publication Number
Snnn	<i>OS/390 MVS System Codes</i>	GC28-1780
ICH IDA IDC IEA IEB IEC IEF IEH IEW IKJ	<i>OS/390 MVS System Messages:</i> <i>Volume 1</i> <i>Volume 2</i> <i>Volume 3</i> <i>Volume 4</i> <i>Volume 5</i>	GC28-1784 GC28-1785 GC28-1786 GC28-1787 GC28-1788
HASP	<i>MVS/ESA JES2 Messages</i>	GC23-0085
IEV	<i>IBM High Level Assembler/MVS Programmer's Guide</i>	SC26-4941
IGZ CEE	<i>LE for OS/390 Debugging and Run-Time Messages Guide</i>	SC28-1942
AFB	<i>IBM VS FORTRAN Version 2: Language and Library Reference</i>	SC26-4221
ILX	<i>VS FORTRAN Version 2: Programming Guide</i>	SC26-4222
IBM	<i>PL/I for MVS and VM Compile-Time Messages and Codes</i>	SC26-3229
ICE	<i>DFSORT Messages, Codes and Diagnosis Guide</i>	SC26-7050
EZF	<i>EZFORM/PTFORM User's Guide</i>	none

5 NORTH – DEFINING AND CONTROLLING A BATCH JOB

Note: Most existing North JCL (job control language) for submitting batch jobs will work on Titan with minimal changes. Some North facilities may no longer work. For information on changes related to the transition, go to:

<http://silk.nih.gov/silk/titan>

5.1 JOB CONTROL LANGUAGE OVERVIEW

Job control language (JCL) is the means by which the user communicates the resource requirements for a job to the operating system and the various components of the OS/390 North System. Through JCL, the user instructs the computer what to do with programs, data sets, and I/O devices.

The major JCL statements of each job are discussed in this section: JOB, JES2, JCLLIB, EXEC, DD, and OUTPUT.

5.2 JOB STATEMENT

[SEE APPENDIX FOR TITAN INFORMATION]

The JOB statement identifies a set of work to the operating system. It contains accounting information as well as parameters for governing the execution of the job.

The JOB statement consists of the characters "/" in positions 1 and 2; followed by four required fields:

- Job name (iiixxxxx),
- Operation (JOB),
- Accounting information, and
- User identification.

An optional fifth field may include any of the parameters described in "Additional JOB Statement Parameters". The format for the standard OS/390 North System JOB statement is as follows:

```
//iiixxxxx JOB (aaa,pppuu,lines,,forms,cpys, ),  
// user identification,additional parameters
```

Job Name Field Parameters (iiixxxxx)

The first field is the job name and consists of three to eight alphanumeric characters. The first three characters (iii) are required and must be the registered user initials. Up to five optional characters (xxxxx) may be used to further identify the job.

Jobs with identical names will execute one at a time in the order they are received. To ensure a particular order of execution, see Section 5.6.1.

Operation Field (JOB)

The second field identifies this JCL statement as a JOB statement. It must always contain the capital letters JOB.

Accounting Information Field Parameters

The accounting information field must be enclosed in parentheses. The first two parameters in this field are mandatory. The remaining five parameters are optional and may be used to override the default values.

aaa	(Required) the user's registered agency code.
pppuu	(Required) the project code for supplemental accounting purposes. If an agency has no requirement for project codes, this field can be used for identifying information.
lines	(Optional) up to three digits to specify the maximum number of lines of printed output, in thousands, directed by the job to SYSOUT data sets.
forms	(Optional) a four-digit form number indicating that the entire job, including JCL listings, is to be printed on a special form. Standard one-part, 12" x 8 1/2" paper will be used if this parameter is omitted. It is recommended that form numbers be specified in the Data Definition (DD) statement. See Section 5.5.2 for details.
cpys	(Optional) up to two digits to specify the number of copies of printed output, including JCL and system messages. If this parameter is omitted, the system will default to one copy. If multiple copies of a specific SYSOUT data set are desired, use either the COPIES parameter on the DD statement or the JES2 OUTPUT control statement.

User Identification Field

myname	(Required) one-to-twenty characters used to identify the person or project group submitting this job. Blanks and/or special characters may be included by enclosing the entire field in apostrophes (e.g., 'MY NAME' or 'TEAM #7')
--------	--

Additional JOB Statement Parameters

The following is a list of optional JOB statement parameters and descriptions of how they apply to the OS/390 North System installation:

ADDRSPC	Not supported by the OS/390 North System.
CLASS	A single letter used to assign a job to a designated job class. See Section 5.2.1 for the available job classes. If this parameter is omitted, the job will default to CLASS=A.
COND	This parameter specifies whether a job will continue to process based on completion codes issued by the job steps. If this parameter is coded, it applies to all steps in the job. If the condition specified is met, all subsequent steps are bypassed.
GROUP	This is a RACF parameter. If a group name is not specified, the default group is the agency code in the accounting field.
MSGCLASS	Specifies the output class to which system messages for your job are to be routed. Codes A, F, T, and X are the only allowable parameters. If this parameter is omitted, the system will default to standard MSGCLASS=A. F causes the output to be produced in standard format on microfiche. T causes the system messages to be queued for online retrieval by a TSO user. X causes all system messages to be deleted.
MSGLEVEL	Indicates what JCL and system messages are to be written as part of the output listing. If this parameter is omitted, the system will default to print all control statements and all allocation/termination messages.
NOTIFY	Specifies that a job completion message be transmitted to a TSO user.
PASSWORD	<ul style="list-style-type: none">• Not supported.
PERFORM	<ul style="list-style-type: none">• Not supported.
PRTY	<ul style="list-style-type: none">• Not supported
RD	<ul style="list-style-type: none">• Not supported.
REGION	Specifies the amount of storage to be allocated to a job and should be used whenever the storage required by a job exceeds or is

	significantly less than the default. If this parameter is not coded, the default region is 512K. Unless a job consists of a single step, the region size should be coded on the EXEC statement rather than on the JOB statement. This parameter's value should be coded as REGION=nnnnK, where nnnn represents multiples of K and K equals 1024 bytes, or REGION=nnnnM, where nnnn represents multiples of M and M equals millions of bytes.
RESTART	Allows restarting at the beginning of a step. The CHECKID subparameter is not supported.
TIME	Specifies the maximum amount of execution CPU time required by the job and should be used whenever the time required by the job exceeds or is significantly less than the default for the job's class. See Section 5.2.1 for the default and maximum values for each job class. This parameter's values should be coded as TIME=(mmm,ss), TIME=(,ss) or TIME=(mmm), where mmm, represents minutes and ss represents seconds. The maximum number of minutes is 998; the maximum number of seconds is 59. The value for TIME= should be estimated as accurately as possible since this parameter determines the job priority within job class. Jobs that exceed the time specified, either by default or in the TIME= parameter, are automatically cancelled.
TYPRUN	Specifies special JES2 processing. Acceptable subparameters are COPY and SCAN. The HOLD and JCLHOLD subparameters are not supported.
USER	Not supported.

Examples of Job Statements

The following examples illustrate the use of the JOB statement, and the use of various parameters to limit the job's use of resources. This should be done whenever possible, particularly when testing new programs.

This following example is a basic JOB statement with default values for job class, lines, and time:

```
//III024XY JOB (AAA,PPPUU),HOLLERITH
```

The following example of a JOB statement can produce up to 10,000 lines of printed output before being terminated, and use up to five minutes of CPU time:

```
//IIILIST JOB (AAA,PPPUU,10),'H HOLLERITH',  
// CLASS=C,TIME=5
```

5.2.1 Job Class Standards

Jobs are divided into classes based on the system resources required for the job. Once queued, jobs are selected according to class and execution time. shows other resource limitations by job class.

Figure 5-1. Resource Limitations by Job Class

Job Class	CPU Time	Max. Tape Units Per Step	Other Limitations
CLASS=A	maximum - 1 min/30 sec default - 1 min/30 sec	5 (NIH)	Processed during attended operations.
CLASS=B	maximum - 1 minute default - 1 minute	1 NIH* and 1 foreign	Foreign tape jobs with one step processing during attended operations. Others, see CLASS=D.
CLASS=C	maximum - 998 minutes default - 10 minutes	5 NIH*	Processed only during attended operations.
CLASS=D	maximum - 998 minutes default - 10 minutes	4 NIH* and 1 foreign OR 3 NIH* and 2 foreign	Processed only during attended operations. Foreign tape jobs with multiple steps or exceeding one minute of CPU time. If two foreign tapes are used, only one can be 6250 bpi.
CLASS=E	maximum - 15 seconds default - 15 seconds	none	No restrictions on processing hours.
CLASS=F	maximum - 998 minutes default - 10 minutes	none	Batch jobs not requiring immediate turnaround, with a 72 hour maximum.
CLASS=R (discount class)	maximum - 998 minutes default - 10 minutes	5 NIH*	Batch jobs not requiring immediate turnaround, with a 72 hour maximum.
CLASS=S (discount class)	maximum - 998 minutes default - 10 minutes	none	Jobs held for unattended weekend and holiday processing; they can be submitted at anytime during the week.

Notes:

* A maximum of two tape drives (UNIT=TAPE) can be allocated in a job step.

The following applies to all job classes:

- Maximum lines per job is 999,000; the default is 50,000.
- Maximum copies printed per job is 99; the default is 1.
- Maximum lines printed per page is 255; the default is 0.

Go to:

<http://silk.nih.gov/public/public.schedule>

for the schedule of operations.

5.3 JCLLIB STATEMENT

The JCLLIB statement is used to define a private procedure library. The OS/390 North System does not provide a public procedure library or permit public access to SYS1.JCLLIB. Users must establish a user-defined private procedure library, preferably at the agency level. The library should be maintained as a partitioned data set (PDS) and all rules for creating and maintaining a PDS apply. The following rules also apply:

- The library must reside on a permanently mounted private or PUBnnn volume.
- The logical record length must be 80 and the blocksize must be a multiple thereof and equal to or less than 4160.

A private procedure library stored on public volumes must be cataloged. The placement of the JCLLIB statement is critical. It must be placed after the JOB statement and before the first EXEC statement. **Note:** the JCLLIB statement requires a space following //.

Examples of JCLLIB Statements

The following JCLLIB statement identifies a cataloged user procedure library:

```
//... JOB ...  
/*PASS ...  
// JCLLIB ORDER=( AAAIII.DSNAME )  
//STEP1 EXEC ...
```

5.4 EXEC STATEMENT

The EXEC statement is the first statement of each job step. It identifies the program to be executed or the cataloged procedure to be invoked. Since a job can have multiple steps, there can be more than one EXEC statement. Through the COND parameter, the user can specify to the operating system whether or not to run a step based on the outcome of a previous step. The user can pass information to the program being run by use of the PARM parameter.

The EXEC statement consists of the characters "/" in positions 1 and 2, followed by four fields:

- Step name (stepname)
- Operation (EXEC)
- Parameters

- Comments

The format for the standard OS/390 North System EXEC statement is shown in the following example:

```
//stepname EXEC parameters comments
```

Step Name Field (stepname)

The stepname field is optional. It must be specified if certain JCL facilities are to be used (i.e., making backward references to the step, overriding parameters in a cataloged procedure, and performing a step restart).

If supplied, the stepname consists of one-to-eight alphanumeric characters and the first character must be alphabetic or an @, #, or \$. If the stepname field is omitted, one or more spaces must separate the // and the keyword EXEC.

Operation Field

The second field identifies this JCL statement as an execute statement. It must always contain the capital letters EXEC.

Parameters Field

At least one parameter, either a program name or a procedure name, must be specified. All other parameters are optional, may appear in any order, and are separated by commas. The parameters are:

PGM	Specifies the program to be executed. This program must be in load module form in a partitioned data set. One-to-eight alphanumeric characters are allowed.
PROC	Specifies the name of a cataloged procedure to be invoked and executed. The keyword PROC= may be omitted; any name that appears in this position not preceded by the keyword PGM= is assumed to be a procedure name.
ACCT	Not supported.
ADDRSPC	Not supported.
COND	Tests condition codes of previous steps to determine if current step is to be executed.
DPRTY	Not supported.
DYNAMNBR	Not supported.

PARM	May be used to pass up to 100 characters of data to the program being executed.
PERFORM	Not supported.
RD	Not supported.
REGION	Specifies the amount of storage to be allocated to the job step. The default is 128K. This parameter has no effect if the REGION= parameter is specified on the JOB statement.
TIME	Specifies the maximum amount of time that a job step may use the CPU; the default is 998 minutes. The total time for all steps may not exceed the time specified in the JOB statement.

Examples of Execute Statements

The following example will execute the program WXYZ and allocate a region of 126976 bytes.

```
//STEPX EXEC PGM=WXYZ,REGION=512K
```

The following example executes the procedure FORTGCL to compile and link edit a FORTRAN program. The TIME parameter will prevent excessive costs in case a runaway condition results from a programming error.

```
//STEPY EXEC FORTGCL,TIME=( , 30 )
```

5.5 DD (DATA DEFINITION) STATEMENT

There must be a DD statement for each data set to be used in a job step. The information included in the DD statement specifies I/O devices, location of the data set, certain data attributes, and special processing options. The DD statements for a job step must follow the EXEC statement for that job step. If DD statements override statements in a procedure, they must appear in the same order as in the procedure. Any additional DD statements for a step follow overriding DD statements for that step.

The DD statement consists of the characters // in positions 1 and 2, followed by four fields:

- data definition name (ddname)
- operation (DD)
- parameters (one or more)
- comments

The format for the standard OS/390 North System DD statement is shown in the following example:

```
//ddname DD parameters comments
```

5.5.1 Data Definition Name Field (ddname)

The ddname field consists of one-to-eight alphanumeric characters and the first character must be alphabetic or an @, #, or \$. The ddname should not be coded when the statement defines a concatenated data set.

The ddname field of the DD statement (not to be confused with the DDNAME= parameter) identifies a DD statement to which other control statements in a program can refer or identifies a DD statement which invokes a particular job control language facility. The ddname begins in position three.

Each ddname within a job step should be unique. If duplicate ddnames exist in a step, all references are directed to the first occurrence in that step.

Certain ddnames are reserved for specific operating system functions. These are:

JOBCAT	Not required.
JOBLIB	Defines a private library made available to the job for searching for a program name. This specification must precede the first EXEC statement.
JCLLIB	Defines a private procedure library.
STEPCAT	Not required.
STEPLIB	Defines a private library made available to a job step for searching for a program name. This specification should follow the execute statement for the job step.
SYSABEND	Defines a data set on which a dump can be written if the job step terminates abnormally. The dump provided would include the processing program area, the system nucleus, and the system queue area. Since it generates large amounts of data to be printed, this statement should not be used unless requested by TASC.
SYSCHK	Not supported.
SYSCHKEOV	Not supported.
SYSMDUMP	Defines a data set on which an unformatted, machine-readable

dump can be written if the job step terminates abnormally.

SYSUDUMP	Defines a data set on which a dump can be written if a job step terminates abnormally. The dump provided would include only the processing program area. This option is sufficient for problem determination in most cases of abnormal termination.
----------	---

Operation Field

The second field identifies this JCL statement as a data definition statement. It must always contain the capital letters DD.

Parameters Field

At least one parameter identifying the type of data set to be processed must be specified. Depending upon the type specified, additional parameters may also be required. The additional parameters may be in any order, separated by commas. See Section 5.5.2 for a guide to the parameters normally required by I/O devices supported by the OS/390 North System.

Following is a list of the data definition statement parameters with a brief description of each. Full descriptions of all parameters may be found in the manual *IBM OS/390 MVS JCL Reference*, GC28-1757.

*	Specifies that data follows in the input stream. For details, see Section 5.5.2.
AMP	Provides for completing information in an access control block (VSAM data sets only).
BURST	Specifies whether output being printed on the IBM 3835 Printing Subsystem is to be trimmed and burst.
CHARS	Selects up to four character sets to be used in printing a data set that is not controlled by print systems defaults, or within a PAGEDDEF.
CHKPT	Not supported by the OS/390 North System.
COPIES	Specifies number of copies to be printed.
DATA	Specifies that data follows in the input stream and that the data contains JCL which is to be treated as data. For details, see Section 5.5.2.
DCB	Specifies data set attributes not defined in the processing program or in data set labels.

DDNAME	Postpones definition of a data set until a DD statement later in the same job step.
DEST	Specifies a destination (either LOCAL or a valid remote number) for routing of output. For details, see Section 5.5.2.
DISP	Specifies creation and disposition information for the data set. For details, see Section 5.5.2.
DLM	Allows specification of a delimiter other than /* to terminate an input stream data set.
DSN	Assigns a name to a data set. For details, see Section 5.5.2.
DUMMY	Specifies that all I/O operations are to be ignored for this data set.
DYNAM	Not supported.
FCB	Specifies the Format Control Block to control vertical spacing for a data set being printed on the IBM 4245 impact printer. This parameter may be used on the IBM 3160 printer when a PAGEDEF is not being used.
FLASH	Not supported.
FREE	Causes deallocation of a data set when it is closed. For details, see Section 5.5.2.
HOLD	Places a SYSOUT data set on the HOLD queue until released by the central, remote, or terminal operator. This parameter must be used for all TSO SYSOUT=T data sets unless MSGCLASS=T is specified in the JOB statement. For details, see Section 5.5.2.
LABEL	Describes the type of label processing to be performed for the data set. For details, see Section 5.5.2.
MODIFY	Not supported.
OUTLIM	Allows specification of a limit for the number of logical records included in a SYSOUT data set.
PROTECT	This RACF parameter is not required since all RACF-defined North System users have Automatic Data set Protection

(ADSP).

QNAME	Not supported.
SPACE	Requests space on a direct access device. For details, see Section 5.5.2.
SUBSYS	Not supported.
SYSOUT	Routes a data set through the system output stream. For details, see Section 5.5.2.
TERM	Not supported.
UCS	Describes the character set to be used for printing an output data set on an impact printer. The OS/390 North System supports the TN character set on the 4245 printers.
UNIT	Specifies the type of device to be used for a data set. For details, see Section 5.5.2.
VOL	Specifies the volume serial number (tape or disk) on which a data set resides or will reside.

5.5.2 DD Parameters for Creating Data sets

The following table shows the parameters generally needed to create a data set using the device specified in the column heading. Some of these parameters may be omitted if the required information is available from another source.

Figure 5-2. DD Parameters for Creating Data sets

Tape	Disk	Printer	COM
DSNAME	DSNAME	SYSOUT=A, C, D, or J	SYSOUT=F
UNIT	UNIT		
DCB	DCB	DCB	
LABEL	SPACE		
DISP	DSP		
	VOL		

*** or DATA (Input Stream Data Set) Parameter**

The * or DATA parameter on a DD statement indicates that input data records immediately follow the DD statement. More than one input stream data set can be coded per job step if the ddnames are unique. The only other parameters that may appear on a DD statement specifying the * are DCB and DLM. The DATA parameter must be specified on the DD

statement when the input records that follow contain JCL statements or data statements that have // in positions 1 and 2.

Input stream data is assumed to be 80-byte logical records. DCB information supplied by the processing program must conform to this rule. If DCB information is not supplied, the characteristics RECFM=F and LRECL=80 will be assumed. Any specification of BLKSIZE and BUFNO will be ignored. The delimiter statement (/*) used to indicate the end of an input stream data set is optional when the * parameter is specified. It is required when the DATA parameter is specified.

```
//ddname DD *  
Input stream data records  
/*
```

An input record containing /* in positions 1 and 2 will terminate the input stream. An input record containing // in positions 1 and 2 will be interpreted as JCL and will terminate the input stream.

```
//ddname DD DATA  
Input stream data records  
/*
```

Input records containing // in positions 1 and 2 will be accepted as data. Only an input record containing /* in positions 1 and 2 will terminate the input stream.

```
//ddname DD DATA,DLM=$$  
Input stream data records  
$$
```

Input records containing anything except \$\$ in positions 1 and 2 will be accepted as data. Only an input record containing \$\$ in positions 1 and 2 will terminate the input stream.

CHARS= (Character Set) Parameter

Specifies fonts to be used when printing. The CHARS parameter is ignored if a font has been specified in a PAGEDEF within an OUTPUT statement. If neither a CHARS parameter nor a PAGEDEF is used, the system default fonts will be used.

DEST= (Output Destination) Parameter

DD statements specifying SYSOUT can also specify a destination for the output. (see Section 5.5.2) The destination can be LOCAL or any valid remote station number. It can be used in conjunction with the JES2 command /*ROUTE, but the DEST parameter on the DD statement will override the destination specified on the /*ROUTE statement for that data set only. Valid designations for remote numbers are Rnnn and RMTnnn where nnn is the 1 to 3 digit remote terminal number. Do not use REMOTEnnn.

DISP= (Data Set Disposition) Parameter

The DISP= parameter describes the status of the data set at the beginning of the job step, the disposition of the data set if the job step terminates normally, and the disposition of the data

set in case of an abnormal termination. Specify DISP=SHR for all public disk data sets and for all other disk data sets involved in the job that are not being updated. The operating system does not force a disposition of CATLG for new generation data group data sets. A disposition of CATLG must be specified to have the new generation appear in the catalog.

DSN= (Data Set Name) Parameter

The DSN= parameter assigns a name to a data set. This parameter is not necessary if the data set is temporary (i.e., if the data set is being created and deleted within the job).

Special data set names are used to designate tapes for large volume print, COM tapes, and tapes that will be purchased. The final disposition of a print, COM, or purchase tape must be KEEP; this will cause the operator to be informed that the tape requires further processing.

OS/390 North System installation standards for data set names for tape or disk data sets, as well as for special purpose output tapes, are specified in Figure 5-3.

Figure 5-3. Data set Naming Standards

Data set Type	Standard Format
Tape and Disk Files	aaa.iihpppuu
TSO Data sets	\$iii.aaapppuu.tttt
TSIO Data sets	TSIO.zzzzzzzz.aaa.pppuu
WYLBUR Data sets	aaa.iihpppuu.wylsuffix
Public Library Members	iihpppuu

Note: See below for details of all fields and characters specified in this figure. In addition to the underscored required fields, user-specified fields can be added up to a total data set name length of 44 characters, including required characters, optional characters, and all periods. For each group of eight characters or less there must be a period, and the character following each period must be alphabetic or an @, #, or \$. For TSO data sets, the additional user-specified fields must precede the tttt data set type field. In all other cases, any additional fields come after the required fields.

Subparameters of DSN=

All fields and characters shown in Figure 5-3 as CAPITAL LETTERS must be entered exactly as shown. The purpose and values of all variable subparameters, required or optional, are:

aaa	Registered agency/organization code.
CPYnnn	Number of copies of printed output or microfiche to be produced. Substitute a one-, two- or three-digit number for "nnn."
iihpppuu	Registered initials (iii) of individual authorized to submit jobs under "aaa," and, optionally, up to five user-specified characters

	(pppuu) for project codes or other data set identification descriptors.
tttt	Must specify a valid TSO data set type qualifier (i.e., DATA, COBOL, CNTL, CLIST, etc.)
wylsuffix	WYLBUR specific data set name extension, LIB, EDIT, CARD, or any qualified suffix.
\$#ac	Controls font selection and indentation.

Examples of DSN=

The following example of a data set name identifies a file created by user III of organization AAA:

```
DSN=AAA.III TRANS
```

In the following example, the data set was created by user III of organization AAA as a TSO file with the optional user descriptor of INPUT. The type designation of DATA was also used. User III included an additional optional descriptor of D091582 to include the date created as part of the data set name.

```
DSN=$III.AAINPUT.D091582.DATA
```

FCB= (Forms Control Buffer) Parameter

Vertical spacing and page ejects are controlled by a Forms Control Buffer (FCB) parameter when printing is directed to the IBM 4245 impact printer. FORMS=x290, which prints on the IBM 3160 also uses the FCB parameter to control spacing and page ejects. Special FCBs to cause printing to start or stop at other than the normal top or bottom line position, or which intermix line spacing increments, must be created and maintained by the user.

The form number/standard FCB combinations are displayed below.

Form Type	Form Size (W & L)	FCB Names
x290	9 1/2" x 11"	S6, S8, or S

The second character of the FCB names listed above is a hexadecimal value representing the spacing increments as follows:

6 = six lines per inch
8 = eight lines per inch

Form Type	Form Size (W & L)	IBM 4245 FCB Names
x010	7 1/2" x 3 1/4" Card	6010, 8010
x050	9" x 5" Card	6050, 8050

x593	14 7/8" x 11"	6593, 8593
x609	1 1/2" x 4" Label	6609, 8609

The first character of each of the IBM 4245 FCB names listed above represents the spacing increment as follows:

6 = six lines per inch
8 = eight lines per inch

The remaining three characters of each IBM 4245 FCB name identifies the form type with which it may be used.

FREE= (Deallocate data set) Parameter

The FREE= parameter causes data set deallocation when the data set specified by the DD statement in which the FREE= parameter appears is closed. This parameter should be used to preclude monopolizing critical resources. The FREE= parameter may be used with all SYSOUT data sets.

The FREE= parameter has two subparameters:

CLOSE	FREE=CLOSE will cause the data set to be deallocated as soon as it is closed. FREE=CLOSE should not be included in a DD statement for a data set that is opened and closed more than once during a job step. FREE=CLOSE is ignored for a DD statement that specifies JOBLIB, STEPLIB, or a data set of a concatenated group.
END	FREE=END will cause the data set to be deallocated at the end of the job step.

HOLD= (Queue for Future Action) Parameter

Specifying HOLD=YES on a SYSOUT data set will place the data set on the HOLD queue until released by the central operator or by a remote terminal operator or user. Output remaining on a SYSOUT= queue for more than 45 days will be purged. The HOLD=YES parameter must be specified for all TSO SYSOUT=T data sets unless MSGCLASS=T is specified in the JOB statement. HOLD= and DEST= can both be specified for the same SYSOUT data set.

LABEL= (Data set Label) Parameter

The LABEL= parameter is used to relate several types of information to the data set label on tapes. The format for the LABEL= parameter is shown in the following example:

LABEL=(seqno,type,password,process,retention info)

Subparameters of LABEL=

The values that may be substituted for each of these subparameters are:

seqno	This is the relative position of a data set on tape. If this subparameter is not specified, the system will supply a default value of 1.	
type	This specifies the label type. There are five possible label types:	
	SL	IBM standard labels. All North System library tapes have IBM standard labels. If a label type is not specified, the system will default to SL.
	AL	American National Standards Institute (ANSI) labels.
	NL	No labels.
	NSL	Not supported.
	BLP	Bypass label processing. This option is not supported. (See Section 6.9.1.5.) BLP will default to NL.
password	Not supported.	
process	Specifies the type of processing for which this data set is being allocated:	
	IN	The data set is to be used for input only. All FORTRAN users must specify IN if the data set is to be used for input only; failure to do so will cause operator intervention and cancellation of the job.
	OUT	The data set is to be used for output only.
	INOUT	The data set is to be used for both input and output.
retention	This keyword parameter supplies information about the retention period or expiration date of a data set. The available retention/expiration keywords and their meanings are:	
	EXPDT=yyyy/ddd	Specifies the date when the data set can be deleted. Except as noted below, the expiration date cannot be more than 365 days

from the creation date.

Example: EXPDT=2000/180 for the 180th day of the year 2000.

EXPDT=98000	Specifies a foreign tape to TMS, and is required to read or write a tape which is not under the control of TMS.
EXPDT=99000	Specifies a tape data set that is to be scratched if and when it becomes uncataloged. This keyword does not apply to disk data sets and will not catalog a disk data set.
RETPD=nnn	Specifies the length of time in days that the data set must be retained. The maximum number of days is 365. Leading zeroes are not required. The default retention period for new tape data sets is five days. RETPD is not recommended for disk data sets.

The following option is to be used for input only. Any attempt to overwrite or delete data from this data set will be denied.

LABEL= (, , , IN)

The following example of the SL option, indicates the data set is the second data set on the volume and has a standard IBM label.

LABEL= (2 , SL)

The following example shows the format for retaining a data set 30 days from the date of creation.

LABEL=RETPD=30

SPACE= (Allocate Direct Access) Parameter

The SPACE= parameter is used to request space on a disk. Space is allocated before the job step begins execution. The requirements and standards for disk data sets are specified in Section 5.5.3. CIT does not guarantee the availability of disk space; however, if space is requested and is unavailable, please contact TASC for assistance.

SYSOUT= (Output Stream Data set) Parameter

The SYSOUT= parameter assigns an output class to an output data set. The following example shows the standard format for the SYSOUT= parameter (class is required) followed by the subparameter options:

`SYSOUT=(class,program,forms)`

Subparameters of SYSOUT=

These subparameters are positional.

class	This subparameter is used to assign the output to an output class.
program	This subparameter is not used; however if the "form" subparameter is used, this missing parameter must be represented by an extra comma.
form	This subparameter specifies a form type or an /*OUTPUT control statement to be used for this output data set. If this subparameter is not specified, the default is RX form (0621).

The output classes established by the OS/390 North System and descriptions of the attributes are listed below:

Class Attributes

A	Print data set, up to one million lines.
J	Print data set, up to one million lines, specifies printer form to be used, and can also invoke upper and lower case printing.
F	Print data set, microfiche media output.
K	Print data set on the IBM 3160 duplex printer.
T	Print data set to be retrieved and disposed of by a TSO user.
X	Data set to be discarded by JES2 without processing.
*	Print data set to use the same output class specified in the MSGCLASS= parameter of the JOB statement.

UNIT= (I/O Device Type) Parameter

This parameter specifies the type of input or output device required to process the data set. The following unit names are used at the OS/390 North System:

FOREIGN	9-track 1600/800 bpi tape unit for tape not under control of TMS. The default output density is 1600 bpi.
FRGN3480	18-track, 38000 bpi cartridge tape unit for cartridge tape not under control of TMS.
FRGN6250	9-track 6250/1600 bpi tape unit for tapes not under control of TMS. The default output density is 6250 bpi.

SYSDA	Direct access device, scratch space.
TAPE	9-track 6250/1600 bpi tape drive.
TAPE9	9-track 6250/1600 bpi tape drive.
CTAPE	18-track 38000 bpi cartridge tape drive.
APL	Direct access device, APL TSIO data sets and workspaces (serial numbers begin with APLVS).
SYSTS	Direct access device, TSO scratch packs (serial numbers begin with SCRTS).
3390	Direct access device. May be used when allocating to specific volumes.
PUB	Direct access device, public online storage space.
WYL	Direct access device, WYLBUR private storage space.

Default or Absent UNIT= Specification

The unit information specified on the JCL statement will override unit information that is available from other sources (i.e., the system catalog or cataloged procedures). Many of OS/390 North System's cataloged procedures contain symbolic parameters that specify a default value for the UNIT= parameter, usually UNIT=SYSDA. If the cataloged procedure's default is not applicable, it must be overridden or nullified.

A typical link edit cataloged procedure specifies on the SYSLMOD output DD statement a temporary data set named "DSN=&&GOSET(GO)" on a direct access scratch space "UNIT=SYSDA". To use a private, cataloged data set, the user's overriding output DD statement should be coded:

```
//LKED.SYSLMOD DD DSN=USER.LOAD(TEST),DISP=SHR,UNIT=
```

By not specifying anything after the equal sign in the UNIT= parameter, the default parameter UNIT=SYSDA in the cataloged procedure is nullified. The system catalog will be searched for data set USER.LOAD to find the correct unit and volume specifications.

OUTPUT JCL Statement

The OUTPUT statement is used to:

- specify the characteristics of the processing options for a specific SYSOUT data set
- group SYSOUT data sets for processing by a printer
- specify default options for output data sets

- set a destination for an output data set

The OUTPUT JCL statement consists of the characters "/" in positions 1 and 2, followed by four fields:

- step name (stepname)
- operation (OUTPUT)
- parameters
- comments

The format for the standard OS/390 North System OUTPUT JCL statement is shown in the following example:

```
//stepname OUTPUT parameters comments
```

Step Name Field (stepname)

Every OUTPUT JCL statement must contain a stepname field. Each stepname must conform to the following:

- Each OUTPUT JCL statement stepname must be unique within the job.
- The stepname must begin in column three.
- The stepname must be one-to-eight alphanumeric or national characters.
- The first character must be alphabetic or national.
- The stepname must be followed by at least one blank.

Parameters Field

There are 29 parameters for use in the OUTPUT JCL statement. Of these, the following are not available at the OS/390 North System:

CKPTLINE	INDEX	TRC
CKPTPAGE	LINDEX	WRITER
CKPTSEC	PRMODE	COMPACT
PRTY	FLASH	THRESHLD

Location in JCL Input Stream

The OUTPUT JCL statement must be placed in the input stream before any SYSOUT DD statement that refers to it. A SYSOUT DD statement can refer to an OUTPUT JCL statement in two ways:

Explicitly	The SYSOUT DD statement contains an OUTPUT parameter that specifies the name of the OUTPUT JCL statement.
------------	---

Implicitly	The SYSOUT DD statement does not contain an OUTPUT parameter. Implicit references are to default OUTPUT JCL statements and require the job to contain one or more OUTPUT JCL statements.
------------	--

If the SYSOUT DD statement does not contain an OUTPUT parameter and the job or step does not contain a default OUTPUT JCL statement, processing of the SYSOUT data set is controlled by the JES2 OUTPUT statement and/or OS/390 North System-defined installation defaults.

5.5.3 Disk Storage and Management

The NIH Computer Center maintains disk devices for the storage of user files as well as for the operating system. The North System supports permanently mounted packs only. The benefits of storing data sets on disk areas follows:

- Several jobs can read the same data set concurrently.
- Data sets can be accessed from teleprocessing systems.
- Comprehensive security for disk data sets is available through the Resource Access Control Facility (RACF).
- There is a faster rate of data transfer.
- Jobs may qualify for a higher priority job class (see Section 5.2.1).
- Operator intervention is not required.

All disk storage space allocations are subject to requirements and limitations. The types of disk storage space provided by the North System and conditions of utilization are described in the following paragraphs.

5.5.3.1 Scratch Packs

Scratch packs are provided for storage of temporary data sets created and deleted within one batch job. Data sets placed on scratch packs must be allocated with the UNIT=SYSDA parameter (see Section 5.5.2), and must be named with a proper temporary data set name.

TSO scratch packs are provided for short-term storage of TSO data sets. Data sets placed on these packs must conform to the TSO data set naming standards as specified in Section 5.5.2 and must be cataloged.

Data sets placed on TSO scratch packs must be allocated with the UNIT=SYSDA parameter (see Section 5.5.2). The retention period for all data sets placed on TSO scratch packs is 14 days from the original date of creation. These data sets are not backed up by the NIH Computer Center.

5.5.3.2 Public Storage User Packs

Public storage packs are provided for long term storage of TSO, WYLBUR, APL, VSAM, and non-TSO data. Data sets placed on these packs:

- must be no more than 9,999 tracks
- must conform to North System naming standards as specified in Section 5.5.2 (DSN parameter).
- must be cataloged

All permanent public storage packs have volume serial numbers of the form PUBnnn, where nnn is a number assigned by the NIH Computer Center. UNIT=PUB should be specified when allocating space on a public pack (see Section 5.5.2). When this parameter is specified, a volume parameter is not required and the system will allocate the requested space on an available volume.

If a data set requires more than 9,999 tracks the user should contact TASC. The request must include number of tracks required, anticipated usage, and expected retention period.

5.5.3.3 Disk Backup

The NIH Computer Center provides backup for data sets stored on public user packs. Data sets created or updated within the previous seven workdays can be recovered directly by using ABR commands. For more information, see Section 5.5.3.4. It may be possible to recover a data set that was deleted within the last four weeks with the assistance of TASC.

5.5.3.4 Space Management of Disk Data sets

Automated Backup And Recovery (ABR) is a disk data set management and archiving software system. ABR accumulates usage statistics for each data set and provides facilities for users to request archiving of data sets to cartridge tape. The North System uses ABR to archive data sets that do not comply with NIH Computer Center standards.

The system accumulates the following information for each disk data set:

- date of last use
- volume serial of pack
- data set name
- data set organization
- creation date
- number of tracks allocated
- number of tracks used

-
- number of unused tracks

Data sets are archived to tape for any of the following reasons:

- nonuse for 75 days
- not cataloged
- nonconformance to data set naming standards as specified in Section 5.5.2.
- exceeding space allocation standards
- user requests

Data sets that have been archived will be retained for 670 days. The following types of data sets cannot be archived:

- data sets with invalid RECFM, LRECL, or BLKSIZE
- data sets on TSO scratch packs or private user packs

Procedures and utilities for archiving and reloading data sets are described below.

Automatic Backup and Recovery (ABR)

The FDR/ABR DASD Management System provides the facility for managing public disk storage. It is composed of three major subsystems—incremental backups, archiving, and DASD management reporting.

Incremental Backup - Incremental backup is the process of making a copy of disk data sets so that they may be recovered if damaged or inadvertently deleted. The backup job automatically selects for backup all data sets that have been modified during the current workday. Users may restore any data set to an earlier generation by reloading it from the backup files. A full volume backup is provided every Saturday and incremental backups are run Sunday through Friday nights.

Archiving - Archiving is the process of removing data sets from Public disk and putting them instead on less expensive compressed DASD and tape. The NIH Computer Center will automatically archive any data set on public online DASD which:

- has not been accessed in 75 days
- does not comply with North System standard naming conventions
- exceeds 9,999 tracks of allocated space

Each night, the archive program is run on the North System on all public storage volumes. Users can invoke the archive process on an individual data set when they feel it is

appropriate. An archived data set will be retained on compressed DASD for 30 days and 3490 cartridge for 670 days.

Note: These time frames run concurrently. After 670 days from archival, the data set is purged from the system.

If an archived data set is referenced by a user (in batch or TSO), ABR automatically recalls the data set. If the backup copy on disk has not expired, ABR will restore the disk backup immediately without operator intervention. If the backup on disk has expired (been archived more than 30 days), ABR will call for the backup tape.

Note: Data sets cannot be reloaded from tape during unattended hours. Users should ensure data sets that need to be accessed in unattended hours are not archived to tape. For the hours of operations, go to:

<http://silk.nih.gov/public/public.schedule>

Archiving is not intended to be a permanent backup for disk files. Data sets that must be held for a period longer than 670 days should be dumped to a NIH Computer Center library cartridge tape with the FDRDSF Utility (see Section 6.6).

DASD Management Reporting - ABR gives the user extensive reporting capabilities. The reports reflect both the status of the DASD volumes and the backups and archives contained within the ABR system. ABR supplies a number of standard report formats, or users may customize their own reports.

ISPF Panels - ABR provides users with panels that operate under the Interactive System Productivity Facility (ISPF) under TSO. These panels give users easy access to most of the functions under ABR. These panels are easy to use and contain complete ISPF tutorial (HELP) panels and messages. The ABR/FDR TSO/ISPF interface is available from the ISPF/PDF Primary Option Menu under "A" for ABR/FDR. The following panels are provided:

- REPORTS - ABR reporting functions
- RESTORE - ABR data set restore
- ARCHIVE - ABR data set archive
- INC-BACKUP - ABR data set backup
- JCL PARMS - Specify FDR JCL and SYSOUT defaults
- FORMAT - Modify format of generated reports
- MESSAGES - FDR messages and codes query facility

- QUERY - FDR/ABR statistics query
- SRS - SEARCH, REPORT, SERVICES DIALOG

ABR CLISTs - Two CLISTs are available to perform ABR functions in line mode through TSO—the LISTARC command that produces an online archive report, and the RESTARC command that submits a batch job that restores archived data sets. The command format is as follows:

```
LISTARC DSN($III.AAA.test.data)
LISTARC DSG($III)
```

- The TSO LISTARC command will display information concerning an individual archived data set. The second will display information about all data sets archived beginning with the prefix \$III.
- The TSO RESTARC command will allow users to restore one or more archived data sets in a single command. Users will be prompted for the name(s) of the data set(s) to be restored if not provided. The format of this command is as follows:

```
RESTARC DSN(dsname,VOL=(volser),NVOL=(new volser))
```

For assistance enter: HELP LISTARC or RESTARC for additional information.

For information on ABR online messages, see Section 7.5.

WYLBUR Commands - WYLBUR users may execute the WYLABRP command to perform ABR reporting functions and WRECALL command to restore an archived data set.

The following example archives information concerning an individual data set:

```
WYLABRP ARCHIVE DSN=AAA.III.test.data
```

The following example displays archive information about all data sets beginning with prefix AAA.III:

```
WYLABRP ARCHIVE DSN=AAA.III.test.data
```

The following example reloads an individual data set from archive:

```
WRECALL DSN=AAA.III.test.data,OVOL=nnnnnn,NVOL=nnnnnn
```

Where: nnnnnn = Original volume location, and NVOL = Optional field to be used to relocate data set to another volume, if desired.

5.6 JOB ENTRY SUBSYSTEM VERSION 2 (JES2)

The Job Entry Subsystem (JES2) is an extension of the Operating System, which controls the processing of all jobs. The OS/390 North System uses the JES2 Multi-Access Spool facility, which allows jobs to execute and print on any OS/390 North System CPU. Jobs are submitted to JES2 from the Web (silkad.nih.gov), from RJE workstations, or from teleprocessing application systems.

When jobs are submitted, JES2 stores them on a direct access device called the "JES2 Spool Disk." Jobs from the spool disk are selected for execution as the required resources become available. All print output is intercepted by JES2 and stored on output spool disks. After job completion, JES2 processes the output when the appropriate output device is available.

This section on JES2 is not a complete reference. It supplements the manual *IBM OS/390 MVS JCL Reference*, GC28-1757, which should be consulted for additional information.

5.6.1 JES2 Job Control Statements

JES2 provides a set of statements which control various aspects of job execution or which provide information to the operating system or to the programs being executed. JES2 control statements contain the characters `/*` in positions 1 and 2 with the command beginning in position 3.

JES2 control statements should be positioned in the job stream after the JOB statement and before the first EXEC statement.

BOX

The BOX control statement is used to direct local job output to an alternate lock box. The format is:

```
/*BOX nnn
```

Where "nnn" is the desired lock box number.

BEFORE/AFTER

The BEFORE and the AFTER control statements may be used to ensure that a series of jobs will be executed in a predetermined sequence. The format is:

```
/*BEFORE jobname
```

or

```
/*AFTER jobname
```

The BEFORE and AFTER control statements are subject to the following restrictions:

- If "jobname" specified in the control statement is not in the system when the control statement is entered, the statement will be treated as a comment
- The "jobname" must be unique or the statement will be treated as a comment
- The "jobname" specified in a BEFORE command must not have begun execution before the control statement is entered, or the statement will be treated as a comment
- Only one AFTER or one BEFORE control statement may be entered with any one job

Examples of BEFORE and AFTER Control Statements

The following example shows how to control job IIIJOB2 from executing until job IIIJOB1 has ended:

```
//IIIJOB1 JOB ...
// EXEC ...

//IIIJOB2 JOB ...
/*AFTER IIIJOB1
// EXEC ...
```

The following example shows how to ensure that job IIIJOB1 will not begin execution until both jobs IIIJOB2 and IIIJOB3 have finished execution.

```
//IIIJOB1 JOB ...
// EXEC ...

//IIIJOB2 JOB ...
/*BEFORE IIIJOB1
// EXEC ...

//IIIJOB3 JOB ...
/*BEFORE IIIJOB1
// EXEC ...
```

In the above example, jobs IIIJOB2 and IIIJOB3 may or may not execute at the same time.

JOBPARM

The JOBPARM statement is a variable field control statement which permits the user to specify job parameters to JES2 via one or more keywords and values. All of the parameters that can be specified in the accounting field of the JOB statement, plus the parameters BURST and RESTART, may be specified as a parameter on the JOBPARM statement.

JOBPARM statement parameter values will supersede the corresponding specific or default value on the JOB statement or on any previous JOBPARM statement. Any number of parameters may be specified on a single JOBPARM statement and any number of JOBPARM statements may be used within a given job. The following example shows the format of the JOBPARM control statement and defines the keyword parameters and values:

```
/*JOBPARM keyword,keyword,...
```

The following table defines the keyword parameter and values of the JOBPARM control statement. Also shown is the short form and their permissible values.

Figure 5-4. JOBPARM Keyword Parameters and Values

Long Form	Short Form	
BURST=y	B=y	Specifies that the output forms printed on the IBM 3835 Printing Subsystem are to be burst and trimmed.
COM=nn	F=nn	Specifies the number of diazo microfiche duplicates to be created when using SYSOUT=F output. The numeric values range from 1 to 99.
COPIES=nnn	N=nnn	Specifies the number of copies to be printed. The numeric values range from 1 to 255.
FORMS=ffff	F=ffff	Specifies form type to be used for printed output. A one-to-four character alphanumeric value.
LINECT=kkk	K=kkk	Maximum number of lines to be printed on each page. The numeric values range from 0 to 255.
LINES=mm	L=mm	Estimated number of output lines to be printed, in thousands. The numeric values range from 0 to 999.
RESTART=y	E=y	Specifies that the job can be restarted from the beginning following a system crash.

NETACCT

The NETACCT control statement is used in jobs which execute on another node, and then transmit SYSOUT to the OS/390 North System for printing. This control statement is required so that the SYSOUT may be associated with its owner at CIT and distributed correctly.

```
/*NETACCT iippiiii
```

Where "iii" is a valid set of OS/390 North System registered initials and "ppppp" is an OS/390 North System project code.

OUTPUT

The OUTPUT statement is a variable-field control statement that permits users to specify different characteristics and/or options for the output data sets within their job. The output characteristics for a given data set are specified as keyword parameters on an OUTPUT statement that is identified by a user-specified code. The code is used to relate the OUTPUT control statement to the output DD statement defining the data set that is to have the specified characteristics.

Parameters specified on the OUTPUT statement will replace any equivalent parameters specified on the related DD statement. If more than one OUTPUT statement specifies the

same "code," the parameters used for the output data sets are unpredictable. Several different DD statements may refer to the same "code."

If an OUTPUT control statement is used for a data set, the third subparameter (form) of the SYSOUT= parameter of the DD statement defining the output data set must specify the same "code" specified by the OUTPUT command. (See Section 5.5.2.) The "code" may be up to four alphanumeric characters. Up to four destinations for SYSOUT data sets can be indicated by enclosing them in parentheses, separated by commas.

Figure 5-5. SYSOUT Form Parameters and Values

Long Form	Short Form	Purpose and Permissible Values
PAGEDEF	PAGEDEF=xxxxxx	Specifies lines per inch, number of lines on a page, print direction, and font selection. Replaced the Forms Control Buffer (FCB) of line printers.
FORMDEF	FORMDEF=xxxxxx	Specifies whether the output will be printed simplex or duplex mode. Can also specify an overlay name created with the Overlay Language (OGL).
BURST=y	B=y	Specifies that the output forms printed on the IBM 3835 printer are to be burst and trimmed.
CHARS=xxxx	X=xxxx	Specifies up to four character sets to be used in printing a data set when not controlled by a PAGEDEF. See Section 5.5.2.
COPIES=nnn	N=nnn	Specifies the number of copies to be printed. Numeric value from 1 to 255.
COPYG=nnn	G=nnn	Specifies the collating arrangement of multiple copies of output printed when not controlled by a PAGEDEF. See <i>IBM OS/390 MVS JCL Reference</i> , GC28-1757.
DEST=ddd	D=ddd	Destination of the output. See Section 5.5.2.
FCB=xxxx	C=xxxx	Forms control buffer for vertical spacing on the IBM 3835 and 4245 printing subsystems. See Section 5.5.2.
FORMS=ffff	F=ffff	Specifies form to be used for printed output.

The following example shows the format of OUTPUT control statements and the related DD statements.

```

/*OUTPUT code keyword,keyword,...

//...JOB...
/*OUTPUT ZZ01 FORMS=3590,N=2,DEST=LOCAL
/*OUTPUT ZZ02 D=(R17,R251),FORMS=RX
...
```

```

      ...
//...EXEC...
//SYSPRINT DD SYSOUT=(J,,ZZ01)
      ...
      ...
//REPORT DD SYSOUT=(J,,ZZ02)

```

- Two copies of the SYSPRINT data set will be printed locally with upper and lower case character set TX12, on 14 7/8" x 11" continuous forms.
- The REPORT data set will be printed at remote terminals R17 and R251 on standard form, RX.

PASS

To prevent unauthorized use of account registration codes, all OS/390 North System users are required to provide a RACF password when submitting a batch job. The /*PASS control statement specifies the current password and, at the users option, a new password. The format is:

```
/*PASS oldpass newpass
```

Where oldpass is the current password and newpass is the replacement. If newpass is not supplied, the current password will continue in force.

Passwords new or old may be any combination of alphanumeric characters from five-to-eight characters long. If both are entered, they must be separated by at least one blank.

The password (oldpass or newpass) must be five-to-eight characters in length and cannot be the same as the userid. When changing passwords, the old and new password cannot be the same.

A password is automatically assigned when a user is registered and can be changed anytime thereafter. There is no provision for deleting passwords.

Jobs submitted online via the TSO or ISPF SUBMIT commands are processed as batch jobs and must include the /*PASS control statement; however, the password should not be recorded in a TSO data set or CLIST. This prevents unauthorized use of the password by persons who may have access to the user's TSO data sets or who may inadvertently acquire printouts of the data sets. When a job is submitted from TSO with the password fields of the /*PASS control statement blank, the system will fill in the current password.

See Section 7.4 for RACF messages.

PRINT

This control statement allows the printing of record(s) the following statement without executing a job. This statement may be placed anywhere within a job and, when encountered, all subsequent input will be immediately queued for printing. Any other JES2 control

statements placed before the PRINT statement have the normal effect. This statement does not eliminate the need for a standard JOB or PASS statement. The format is:

```
/*PRINT
```

Printing may be routed to a site other than the initiating location by preceding the /*PRINT command with a /*ROUTE control statement. Listings are in 80/80 format. All JCL following the /*PRINT command will be printed until an end of file condition is encountered.

This function may also be performed by specifying TYPRUN=COPY and MSGCLASS=A on the JOB statement. See Section 5.2 for details.

PRIVATE

This control statement will cause the word "PRIVATE" to be printed on every line of the header and trailer pages of the job's output printout. When microfiche output is specified, this command will cause the symbol "PRI" to appear in the upper left corner of the title section of all fiche produced by the job. The format is:

```
/*PRIVATE
```

This statement should be used for all printed reports and microfiche that contain personal, proprietary, or other sensitive data. Distribution of all locally produced reports and microfiche produced with this designation are controlled by the OS/390 North System. Contact TASC for more information.

ROUTE

The ROUTE control statement is used either to direct the job execution to specific resources or to direct print or punch output to a location different from the submitting location. Specifying a resource on the ROUTE XEQ statement automatically assigns the job to the proper CPU. Jobs that do not include a ROUTE statement will be cancelled if executed on a CPU which does not have access to a required resource. Up to four resources may be specified when a combination of software and hardware dependencies is required. However if a combination of resources is requested that cannot be resolved together, the job will fail with an appropriate error message. The format of the ROUTE statement is shown in the following examples provide the format for the ROUTE statement:

```
/*ROUTE XEQ resource
or
/*ROUTE PRINT destination
or
/*ROUTE PUNCH destination
```

Parameters

resource

A single resource is specified on each statement. When a combination of resources is required, code multiple Route statements with the resource on the first one being "MULTI".

Some of the resources and their purposes are:

ADABAS	CPSC ADABAS
ADAOFM	the OS/390 North System ADABAS
HOLDADA	after ADAOFM backups
CICS	CICS software
HOLDCICS	when CICS is down
HOLD12AM	after 12 a.m.
HOLD2AM	after 2 a.m.
HOLD4AM	after 4 a.m.
HOLDTP	after 10 p.m.
M204	MODEL 204
HOLD204	after MODEL 204 backups
SASSHARE	SASSHARE server available
destination	This specifies the default destination of all print or punch output data sets produced by the job. This specification can be overridden for any particular output data set by specifying a destination on the output DD statement defining it. See Section 5.5.2 for details (DEST).

RJE workstations do not need a `/*ROUTE` statement. All output is automatically routed back to the site that submitted the job unless otherwise specified.

This job will run on the CPU after the ADAOFM backups are complete. If multiple `ROUTE` statements are coded and "MULTI" is not indicated, then only the last resource coded is used.

```
/*ROUTE XEQ MULTI
/*ROUTE XEQ HOLDADA
```

This statement will direct the print output to remote terminal ID number 23.

```
/*ROUTE PRINT RMT23
```

The job containing this ROUTE control statement updates a MODEL 204 database. This statement will cause the job to be held for processing until MODEL 204 backup has completed.

```
/*ROUTE XEQ HOLD204
```

XMIT

The XMIT control statement is used to identify a job which will be executed at a node other than the originating node. A job card image for the originating site must precede the XMIT control statement, and a job card image for the executing site must follow the XMIT control statement. The format is:

```
/*XMIT nodename DLM=xx
```

where "nodename" is the name of the node to which the job is to be transmitted, and "xx" is some delimiter such as "\$\$" which is coded in position 1-2 of the card image that terminates the statements to be transmitted. If the DLM=xx parameter is not coded on the XMIT statement, the default delimiter is:

```
"/*
```

5.6.2 JES2 RJE Control Statements

SIGNON

The SIGNON control statement must be the first statement entered at the RJE work station after communication has been established with the central computer. The following example shows the format of the SIGNON control statement:

```
/*SIGNON PCCRJERMTnnn password
```

Parameters:

nnn	The remote terminal work station number.
password	The password is assigned by CIT. If the RJE operator wishes to change the password, the help desk must be notified and a new password will be assigned.

SIGNOFF

The SIGNOFF control statement should be used at an RJE work station to notify the central computer that all processing has been completed. Thirty seconds after issuing this command, the telephone will be disconnected and connect charges will cease. The format of the SIGNOFF control statement is:

```
/*SIGNOFF
```

5.7 JOB SUBMISSION

The following section describes the methods of submitting jobs to the OS/390 North System.

5.7.1 Remote Job Entry (RJE)

The OS/390 North System supports high-speed remote job entry (RJE). Workstations located at user sites communicate with the central facility via telephone lines.

5.7.2 Interactive Job Submission

The OS/390 North System supports several interactive teleprocessing systems (e.g., TSO, WYLBUR, and Model 204), which include capabilities for job submission via remote terminals. For details on submitting batch jobs, refer to the specific documentation for the teleprocessing system to be used.

5.7.3 Web Job Submission

North System batch jobs can be submitted through the World Wide Web by going to:

`silkad.nih.gov/tsocmd`

and entering a data set name or executing a CLIST that will submit a job.

5.7.4 Batch Job Cancellation

There may be instances when a user wishes to cancel a batch job after it has been submitted for processing. To cancel a batch job, use one of the following procedures:

- High-Speed Remote Job Entry (RJE) - RJE operators can cancel batch jobs submitted from, or routed to, their work station by entering a properly coded JES2 Control Statement.
- Interactive - Jobs submitted through interactive teleprocessing systems (i.e., TSO or WYLBUR) may be cancelled if the particular teleprocessing system involved provides commands for that purpose. Refer to the documentation for the particular teleprocessing system for further details.
- Web - Go to:

`silkad.nih.gov/tsocmd`

and select a command from the pull-down menu.

Note: If a batch job is cancelled while executing, the job will continue to execute until completion; however, the output will be purged. The user will be billed for job execution even though there is no output.

6 NORTH - SERVICE AND UTILITY PROGRAMS AND PROCEDURES

This section lists the service and utility programs and procedures offered by the OS/390 North System.

Most North System cataloged procedures will be available on Titan, so it is expected that most users' jobs will run on Titan with little or no change. For more information on Titan, go to:

<http://silk.nih.gov/silk/titan>

Figure 6-1. Directory of Utilities (N)

DMBATCH, 308	MARKREST, 326
FDRDSF, 298	MARKUTIL, 326
FRGNCOPY, 320	MAX BATCH, 294
IEBCOPY, 293	MAX DATA/UTIL, 294
IEBDG, 293	MAX/PDF, 294
IEBGENER, 293	Online SyncSort, 298
IEBTPPCH, 293	PCCDCODE, 315
IEBUPDTE, 293	PCCNCODE, 315
IEHPROGM, 293	PCCVTOC, 301
MARKCON, 326	RCVFILE, 304
MAKDUMP, 326	SANITIZE, 324
MARKIV, 326	SENDFILE, 304
MARKIVD, 326	SYNCSORT, 296

6.1 IBM UTILITIES

Utilities are a group of batch programs that can be used to perform a variety of common functions. For further information concerning batch IBM utility programs, contact TASC or consult the manual *IBM DFSMS/MVS Utilities*, SC26-4926

IEBCOPY	Copies or merges partitioned data sets, selects or excludes specified members in a copy operation, and renames and/or replaces selected members of a partitioned data set.
IEBDG	Creates test data sets consisting of patterns of data. This utility can be used to read and write partitioned and sequential data sets.

IEBGENER	Copies records from a sequential data set or a member of a partitioned data set and creates either a sequential or partitioned data set.
IEBPTPCH	Prints or punches a data set. Not supported by the OS/390 North System for local output.
IEBUPDTE	Updates sequential or partitioned data sets.
IEHPROGM	Scratches data sets or members, renames data sets, and catalogs and uncatalogs data sets.

6.2 ACCESS METHOD SERVICES

Access Method Services functions are invoked in batch mode by executing PGM=IDCAMS. The most commonly used functions at the OS/390 North System are listed below:

- defining the base catalog entry for a Generation Data Group
- defining VSAM files
- modifying VSAM file attributes
- loading/unloading VSAM file data
- defining and building alternate indexes for VSAM files
- deleting VSAM files
- printing VSAM files
- checking the structural integrity of VSAM files and indexes

For addition information about IDCAMS, refer to IBM's manual *DFSMS/MVS Access Method Services for Integrated Catalog Facility*, SC26-4906.

6.3 MAX

MAX is a suite of three software products designed for data manipulation in the OS/390 environment. The individual products are named MAX DATA/UTIL, MAX/PDF, and MAX/BATCH.

MAX DATA/UTIL

This product is a data utility that functions exclusively in the area of VSAM and SAM (Sequential Access Method) data sets. The SAM portion came into being to fill the void created by ISPF/PDF's inability to edit records whose length was over 256 bytes. Even though IBM rectified that situation in Release 4.0 of ISPF, MAX DATA/UTIL still contains many useful SAM functions that have no counterpart in ISPF. Since ISPF/PDF still has no

capacity in the VSAM area, software such as DATA/UTIL is a definite requirement for working with VSAM data sets.

In addition to providing browsing and editing capability in the VSAM and SAM areas, MAX DATA/UTIL provides the ability to select a subset of records from a file for viewing and updating. It also provides the means to selectively copy records from an input file to an output file or to multiple output files based on the content of each record. DATA/UTIL also has the functionality to reformat records on output; changing the size of the record if required.

DATA/UTIL provides online access to the IBM utility IDCAMS and its DELETE, DEFINE, ALTER, REPRO, BLDINDEX and VERIFY functions. It allows an existing VSAM data set to be used as a model for creating new data sets; adopting the parameters of the model but allowing individual parameters to be changed. It comes with an extensive help system that is sensitive to the cursor position on the screen.

MAX BATCH

MAX BATCH performs most of the features that the online MAX DATA/UTIL does but in the batch mode. In addition to VSAM and SAM, PDS capability is added in the batch mode. Simple, easy-to-use control statements permit the selective copying, updating and reformatting of data sets. The batch mode allows the product to be incorporated into production job streams to select out records for further processing by subsequent job steps.

MAX/PDF

MAX/PDF is designed to enhance the ISPF/PDF Browse and Edit functions and expands on the data set information available in a VTOC listing. MAX allows the building of data set name lists (DSNLs) that enable the user to group together all of the data sets associated with a given project or particular task. These DSNLs are similar to the lists formed by ISPF/PDF option 3.4. MAX differs in that the user can tailor each DSNL to contain any specified data set name. The user is not confined to a list matching just one partially qualified DSN as with ISPF option 3.4. The DSNs on a DSNL are numbered and can be selected from the command line without the need to move the cursor to the particular line. This simple facility saves considerable time and annoyance. In addition to being able to select the DSN by number the user can also supply a letter to indicate what they want to do with the DSN; e for edit, b for browse, etc.

Each user can have as many DSNLs as they wish and working from these lists is a great saver of time and effort. The keystrokes required to do a quantity of work are greatly reduced.

One useful feature of MAX/PDF is the ability to reduce the Member Selection List (MSL) of a PDS to one that contains only those members containing a user provided search criteria. For example, while looking at a MSL of a PDS, the user can reduce the MSL to only those members that contain the string "FORTRAN" by issuing a simple Find command. After reducing the MSL to those members containing the criteria the members can be browsed or

edited or the whole reduced MSL can be bulk edited or copied to another data set via simple commands.

Another useful feature of MAX/PDF is called "Point n Shoot." By placing the cursor in a data set name occurring anywhere on the screen and pressing a PF key, the users session is automatically transferred to a screen that has that data set name inscribed. The user can then edit, browse or obtain information about the characteristics of that data set. When finished, the session is returned to where the "Point n Shoot" was initiated. This feature allows a programmer to answer questions about data sets without interrupting their session and chain of thought.

For more information on MAX, go to:

<http://silkad.nih.gov/public/pcc.ejs.@www.max.htm>

6.4 SYNCSORT OS/VS UTILITY (SORT/MERGE/COPY)

[NOT SUPPORTED ON TITAN]

SYNCSORT is a high performance SORT/MERGE/COPY utility designed for use on IBM systems that executes under the OS/390 environment. SYNCSORT is a batch utility program that provides three major functions:

- sorting – recalling data set records to produce a specific sequence
- merging – combining up to 100 pre-sequenced data sets into one data set that has the same sequence
- copying – reproducing a data set without going through the sorting process

The SORT/MERGE utility automatically chooses the most efficient sorting or merging technique based on information supplied by the specific sorting application. The input records may be fixed or variable length, blocked or unblocked, tape, or disk data sets.

Workfiles adequate to the size of the sorting job must be specified by the user and may be tape or disk. However, CIT strongly recommends the use of disk workfiles. To compute the number of cylinders of disk workspace required:

- Multiply the record length by the number of records to be sorted.
- Divide the product by 1,560,000 and round the result up to the nearest whole number.

The following example of workspace calculation shows a data set consisting of approximately 75,000 records to be sorted with each record being 80-characters long.

Example 147

```

75,000 x 80
-----
1,560,000      = 3.85 = CYLS=4

//STEPNAME EXEC DISKSORT,CYLS=n
//SORTIN DD ...(Input data set specification.)
//      DD ...(Optional. - Additional input concatenations)
//      ...
//      ...
//SORTOUT DD ...(Output data set specification.)
//SYSIN DD *
        (SORT/MERGE Control Statements)
//      ...
//      ...
/*
```

In addition to workfile specifications, the user must supply the following DD statements:

SORTIN	Describes one or more input data sets. Multiple input data sets are specified by concatenation.
SORTOUT	Describes the output data set which may be any output device or media supported by the OS/390 North System.
SYSIN	This identifies an input stream or other data set which contains properly formatted SyncSort control statements.

Refer to the *SyncSort MVS Programmer's Guide*, available from the CIT Technical Information Office, for the control statement parameters and for details of other functions and facilities provided by this utility.

Three cataloged procedures are provided – SYNCSORT, DISKSORT, and SORT:

SYNCSORT	SYNCSORT is a high performance SORT/MERGE/COPY utility designed, for use on IBM systems, which executes under the OS/390 environment.
DISKSORT	This procedure, which is recommended by CIT, provides DD statements allocating disk workfiles and includes a keyword parameter through which the user specifies the required space according to the formula stated above. The example above shows the format for the DISKSORT procedure. The keyword parameter is described below.

SORT	This procedure does not provide any workfile allocations and is intended for use only when DISKSORT cannot be used. To use the SORT procedure, the user must provide workfile DD statements in addition to the SORTIN, SORTOUT, and SYSIN DD statements specified above. No keyword parameters are provided for the SORT procedure.
------	---

Up to 31 additional input data sets may be concatenated to the SORTIN DD statement. SORTIN may be random-sequenced records for a sort operation or may be previously sorted data sets for a merge operation.

Keyword Parameter

CYLS=n	Optional - Parameter "n" may be one-to-three digits representing the product of the workspace computation specified above, rounded up to the next whole number. If this keyword is not specified, a default of CYLS=50 is used. This will allow sorting of up to 190,000 80-character records.
--------	--

6.5 ONLINE SYNCSORT (OLS)

The Online SyncSort facility allows the user a quick and easy method to build and save SyncSort JCL without knowing the intricacies of the SyncSort environment and language. You do, however, need to know what your input record fields look like, their starting positions, what you want your output records to look like, and where the fields are to be placed. The OLS facility will prompt you for the information from user-friendly panels, with help screens, if needed. The system will also allow you to merge files and format your print output with titles, headers and trailers. You provide the information and OLS will generate all the necessary JCL and storage control cards. To access OLS from the North System, enter P.13 from ISPF's primary option menu, or type OLS from ISPF option 6.

6.6 FDRDSF (FAST DUMP/RESTORE)

FDRDSF (Fast Dump Restore - Data set Function) is a batch utility that creates cartridge tape backup of disk data sets or restores previously dumped data sets from tape. An unlimited number of data sets can be dumped or restored in a single run.

FDRDSF dumps from DASD to cartridge tape by data set name. For sequential or partitioned data sets, only the space actually used is dumped. For direct access data sets, all allocated space is dumped. If multiple data sets are dumped during a run, FDRDSF can later restore the data sets selectively.

This utility does allocate space on a restore function. However when restoring, sufficient primary space must be present to receive the data set. The data set can be restored to the same volume or to another volume. The data set attributes are copied to tape during a dump; this information will be brought back along with the data during a restore.

FDRDSF processes generation data sets by generation number through the use of a control statement. The DSNAME field of the DD statement is coded with the appropriate generation data group name and generation data set level number.

The following example shows the JCL requirements for FDRDSF.

Example 148

```
//STEPNAME EXEC PGM=FDRDSF,REGION=4M
//SYSPRINT DD SYSOUT=A
//TAPE1 DD ... (The cartridge tape being processed.)
//DISK1 DD ... (The disk being processed.)
//DD1 DD ... (Generation group.
//SYSIN DD *
(Control Statements)
/*
```

Control Statements:

DUMP DSN=dsname or DD=ddname

RESTORE DSN=dsname,NEWNAME=dsname or DD=ddname

Control Statement Parameters:

DSN=dsname	specifies the data set to be dumped from the disk or restored from cartridge tape.
NEWNAME=dsname	specifies the new name of the data set.
DD=ddname	specifies a DD statement which contains the required data set name. This statement is required for data generation groups only.

Additional Information:

- Only the unit, volume serial number, and disposition parameters are required on the DISK1 DD statement.
- The DD1 DD statement is required to dump or restore data generation groups or to allocate new space. If multiple data generation groups are dumped or restored, this DD statement may be specified as DD2, DD3, DD4, etc.

The following example dumps a data set (AAA.III.MYDATA) from PUB111 to a cartridge tape that is to be cataloged and retained for thirty days.

Example 149

```
//STEPNAME EXEC PGM=FDRDSF,REGION=4M
//SYSPRINT DD SYSOUT=A
//TAPE1 DD DSN=AAA.III.MYDATA.BACKUP,DISP=(,CATLG),
// UNIT=CTAPE,LABEL=RETPD=30
//DISK1 DD UNIT=3390,VOL=SER=PUB111,DISP=SHR
//SYSIN DD *
        DUMP DSN=AAA.III.MYDATA
/*
```

The following example restores the data set dumped in the previous example to PUB111. The tape data set was cataloged when created. Disk space has been allocated in advance.

Example 150

```
//STEPNAME EXEC PGM=FDRDSF,REGION=4M
//SYSPRINT DD SYSOUT=A
//TAPE1 DD DSN=AAA.III.MYDATA.BACKUP,DISP=OLD
//DISK1 DD UNIT=3390,VOL=SER=PUB111,DISP=SHR
//SYSIN DD *
        RESTORE DSN=AAA.III.MYDATA
/*
```

The following example restores the data set AAA.III.MYDATA and changes the name to AAA.III.MYDATA.OLD.

Example 151

```
//STEPNAME EXEC PGM=FDRDSF,REGION=4M
//SYSPRINT DD SYSOUT=A
//TAPE1 DD DSN=AAA.III.MYDATA.BACKUP,DISP=OLD
//DISK1 DD UNIT=3390,VOL=SER=PUB111,DISP=SHR
//SYSIN DD *
        RESTORE DSN=AAA.III.MYDATA,
        NEWNAME=AAA.III.MYDATA.OLD
/*
```

The following example restores the latest version of a generation data group to PUB111.

Example 152

```
//STEPNAME EXEC PGM=FDRDSF,REGION=4M
//SYSPRINT DD SYSOUT=A
//TAPE1 DD DSN=AAA.III.DATA,DISP=OLD,VOL=SER=nnnnnn
//DISK1 DD UNIT=3390,VOL=SER=PUB111,DISP=SHR
//DD1 DD DSN=AAA.III.DATA(0),DISP=OLD
//SYSIN DD *
        RESTORE DD=DD1
/*
```

6.7 PCCVTOC

[NOT SUPPORTED UNDER TITAN]

The PCCVTOC utility is provided to display the volume table of contents (VTOC) of one or more online disk packs. This utility is provided as a cataloged procedure for batch operation and as a TSO command for interactive use.

The complete volume table of contents (VTOC) of the PUB111 disk pack will be listed in an abbreviated 80-column wide report format in the following example.

Example 153

```
//STEPNAME EXEC PCCVTOC,PARM='EVOL=PUB111,*'
```

An alternative to the PCCVTOC utility is the SHOVRTOC utility, which is also a TSO command. In batch, it is used as follows:

Example 154

```
//stepname EXEC BATCHTSO,SYSOUT='*'  
//DD1 DD UNIT=3390,VOL=SER=PUB111,DISP=SHR  
//SYSPRINT DD *  
//SYSIN DD *  
SHOWVTOC PUB111 PR(ALL)
```

6.8 HOST-TO-HOST FILE TRANSFER

Host-to host file transfer allows files to be transferred between host computer systems.

6.8.1 Network Job Entry (NJE)

[SEE APPENDIX SECTION 8.3 FOR CURRENT TITAN INFORMATION]

The OS/390 North System provides a capability whereby a batch job can execute at one data center and output can be printed at another. This facility is easy to use and does not require significant JCL changes to implement. This service is most valuable to those who are depending upon courier service to transport their printed output to other data center locations.

The node names for the data centers are as follows:

Node Name	Data Processing Center
PCCJES2	NIH North System
BETHJES2	CDSI
BFD	FDA Center for Food Safety
CDCJES2	Center for Disease Control
FDACFSAN	FDA Center for Food Safety
FSAJES2	Family Support Administration

HCFACOM	Administration for Children and Families
HCFJES	Administration for Children and Families
HDSHQ	HDS
HCFADOM	HCFA
NCHSJES2	NCHS at Research Triangle Park
NCHSPSDB	NCHS at Research Triangle Park
SSAPRD1	Social Security Administration
NIHCU	NIH South System
USCSFRN	U.S. Customs
SSAPRD3	Social Security Administration

The following example shows the JCL used to submit jobs from the NIH OS/390 North System to the NIH OS/390 South System for execution. The output is then routed to the NIH OS/390 North System.

Example 155

```
//iiiBR14 JOB (aaa,pppuu,1,0),PCCSUBMIT,
// CLASS=E,MSGCLASS=T,NOTIFY=$iii
/*NETACCT iiippuu
/*XMIT NIHCU DLM=$$
//jjjTEST JOB (bbbb,xxx,A,5,2,0),NIHJOBRUN
/*KEYWORD=kkk
//ASTEP EXEC PGM=IEFBR14,REGION=20K
/*
$$
```

Where:

iii =	North System registered initials
aaa =	North System agency code
pppuu =	North System project code
jjj =	South System registered initials
bbbb =	South System account code
xxx =	South System box number
kkk =	South System keyword

The NETACCT statement must accompany each job that is submitted from the OS/390 North System. This statement is required so that the North System can identify the SYSOUT data sets that are returned to the North System for printing. If this statement does not appear in the job stream JCL, the output returned from the South System can not be identified and will be discarded.

XMIT is the statement that identifies the job as one to be executed at a site other than the originating site. This parameter specifies the NODE name for the receiving site. Unless specified to the contrary, a job output is always returned to the originating site for printing. If a job is submitted from an RJE station attached to the North System, the RJE station is the originating site and the printing will occur there. The following example shows the JCL used to submit a job from the South System (through WYLBUR or TSO) for execution on the South System, with printed output on the North System REMOTE 2.

Example 156

```
//jjjTEST JOB (bbbb,xxx,A,5,2,0),'your name '  
/*KEYWORD=kkk  
/*NETACCT iiipppuu  
/*ROUTE PRINT PCCJES2.R2  
// EXEC PGM=IEBGENER,REGION=512K  
//SYSPRINT DD SYSOUT=A  
//SYSUT1 DD *  
ABCDEFGG  
ABCDEFGG  
ABCDEFGG  
ABCDEFGG  
ABCDEFGG  
ABCDEFGG  
ABCDEFGG  
/*  
//SYSUT2 DD SYSOUT=A  
//SYSIN DD DUMMY
```

Where:

- iii = North System registered initials
- aaa = North System agency code
- pppuu = North System project code
- jjj = South System registered initials
- bbbb = South System account code
- xxx = South System box number
- kkk = South System keyword

The NETACCT statement in this job stream defines the job to print at the North System. If this parameter is incorrectly coded the SYSOUT data sets cannot be identified and will be discarded. The ROUTE statement instructs JES2 at the OS/390 South System to return all printed output produced by JOB jjjTEST to RMT2 which is attached to the OS/390 North System .

Judicious use of the XMIT and ROUTE statements allow full routing capabilities for execution and printing of jobs at any of the NODE named data centers.

Note: Cannot precede /*XMIT statement with /*ROUTE PRINT statement.

For a job that executed at the North System to print at the South System, the following two statements must be included:

```
/*NETACCT aaaaiii  
/*NETPASS kkkbbb
```

Where:

- aaaa = South System account code
- iii = South System registered initials
- kkk = South System keyword

bbb= South System box number

If a remote is specified, leave the box number off the NETPASS statement.

6.8.2 File Transfer Facility (SENDFILE, RCVFILE)

There are two programs available for electronically transferring any size data sets between data processing centers. The names of the programs are SENDFILE and RCVFILE.

SENDFILE

This program reads the data set to be transferred and converts it into 80-byte records for transmission, adding the necessary control information to permit the data set to be reconstructed at the receiving site. It also accepts a job stream to be executed at the receiving site to reconstruct the data set.

Required DD statements:

SYSUT1	defines the sequential data set to be transferred. The only restriction on this data set is that variable-length spanned records cannot exceed 32,760 bytes in length. Note that it is possible to transfer non-sequential data sets by first creating a sequential copy (e.g., an IEBCOPY unload of a partitioned data set, an AMS EXPORT, or REPRO of a VSAM data set.)
SYSUT2	defines the output containing the job stream and converted data set.
JCL	defines a data set containing a job stream (to be executed at the receiving site) that will be included with the transmitted data. The job stream must include a RCVFILE step to reconstruct the original data set. SENDFILE does not examine the JCL data except to look for the first occurrence of a record containing **SENDFILE** in positions 1-12. This record tells SENDFILE to insert the data set being transmitted at this point. The **SENDFILE** record is only required when there is a step which follows the RCVFILE step.

Completion Codes: SENDFILE returns a completion code of 4 if SYSUT1 defines a null data set, and 0 if it does not. The program issues a U0100 ABEND if a DD statement is omitted.

RCVFILE

This program reconstructs a data set from the 80-byte records created by SENDFILE.

Required DD statements:

SYSUT1	defines the data records created by SENDFILE.
--------	---

SYSUT2	defines the reconstructed data set. The RECFM, LRECL, and, if not overridden, BLKSIZE will be those of the original data set. If a new BLKSIZE is not specified, the following applies: <ul style="list-style-type: none"> • For fixed-length records, if BLKSIZE is not equal to LRECL the "blocked" indicator will be set. • For variable-length records, if BLKSIZE is less than LRECL+4, the "spanned" indicator will be set; otherwise, the "spanned" indicator will be turned off. • For undefined length records, the new BLKSIZE must be at least as large as the original BLKSIZE; otherwise a S013-34 ABEND will occur.
SYSPRINT	defines a message data set where one of several possible messages will be written.

Completion Codes: RCVFILE returns a completion code of 0 if the data set is successfully reconstructed, 4 if SYSUT1 defines a null data set and 12 if an error occurs that forces RCVFILE to terminate. The SYSPRINT message indicates the execution result. RCVFILE issues a U0100 abnormal termination if a DD statement is omitted.

File Transfer From the OS/390 North System To Another Data Center

The following example illustrates how to do a file transfer from the North System to another data center.

Example 157

```
//iiisEND1 JOB
(aaa,pppuu,1,0),FILETRAN,CLASS=E,MSGCLASS=T,
//  NOTIFY=$iii
/*PASS      wwwwww
//STEP1     EXEC  PGM=SENDFILE
//SYSUT1    DD   DISP=SHR,DSN=dataset name
//SYSUT2    DD   SYSOUT=(A,INTRDR)
//JCL       DD   DATA,DLM=##
//iiixMIT1 JOB (aaa,pppuu,1,0),FILETRAN,CLASS=E,
/*PASS      wwwwww
/*NETACCT iiipppuu
/*XMIT      nnnnnnnn
(job statement for receiving data center)
(any other statements required by receiver)
//RECEIVE   EXEC  PGM=RCVFILE
//SYSPRINT DD   SYSOUT=*
//SYSUT2    DD   (parameters defining reconstructed data
set)
//SYSUT1    DD   *
##
```

Where: iii North System registered initials

aaa	North System agency code
pppuu	North System project code
wwwww	North System password
nnnnnnn	Node name for desired data center

File Transfer From Another Center To The OS/390 North System

The following example illustrates how to transfer a file from another center to the North System.

Example 158

```
(job statement for sending data center)
(any other control statements required by sender)
//SEND      EXEC  PGM=SENDFILE
//SYSUT1    DD   DISP=SHR,DSN=dataset name
//SYSUT2    DD   SYSOUT=(A,INTRDR)
//JCL       DD   DATA,DLM=##
(job statement for sending data center)
(any other control statements required by sender)
/*XMIT PCCJES2
//iiiSEND1 JOB
(aaa,pppuu,1,0),FILETRAN,CLASS=E,MSGCLASS=T,
//  NOTIFY=$iii
/*PASS      wwwwww
//RECEIVE   EXEC  PGM=RCVFILE
//SYSPRINT DD   SYSOUT=*
//SYSUT2    DD   (parameters defining reconstructed data
set)
//SYSUT1    DD   *
**SENDFILE**
##
```

File Transfer Of An Entire PDS

The following example illustrates how to transfer a entire PDS.

Example 159

```
//iiiSEND1 JOB (aaa,pppuu,1,0),FILETRAN,
// CLASS=E,MSGCLASS=T,
// NOTIFY=$iii
/*PASS      wwwwww
//UNLOAD    EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//IN        DD DISP=SHR,DSN=dataset name
//OUT       DD DSN=&&TEMP,
// DISP=(,PASS),UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSIN     DD *COPY INDD=IN,OUTDD=OUT
/*
//SEND      EXEC PGM=SENDFILE,COND=(0,NE)
//SYSUT1    DD DISP=(OLD,DELETE),DSN=&&TEMP
//SYSUT2    DD SYSOUT=(A,INTRDR)
//JCL       DD DATA,DLM=##
//iiixMIT1 JOB (aaa,pppuu,1,0),FILETRAN,CLASS=E
/*PASS      wwwwww
/*NETACCT iiipppuu
/*XMIT      nnnnnnnnn DLM=YY
(job statement for receiving data center)
(any other statements required by receiver)
//RECEIVE   EXEC PGM=RCVFILE
//SYSPRINT DD SYSOUT=*
//SYSUT2    DD DSN=&&TEMP,DISP=(,PASS),
// UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT1    DD *
**SENDFILE**
/*
//LOAD      EXEC PGM=IEBCOPY,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//IN        DD DISP=(OLD,DELETE),DSN=&&TEMP
//OUT       DD
(parameters defining reconstructed data set)
//SYSIN     DD *
              COPY INDD=IN,OUTDD=OUT
/*
YY
##
```

6.8.3 CONNECT:Direct (Network Data Mover)

[SEE APPENDIX FOR CURRENT TITAN INFORMATION]

CONNECT:Direct, also known as Network Data Mover (NDM), allows host-to-host file transfer of financial data.

The following example illustrates how to sign on to the NDM using the batch interface program (DMBATCH). Change the control cards to reflect your request requirements.

Example 160

```
//iiBATD JOB (aaa,pppuu),NDM,CLASS=G,
// MSGCLASS=T,NOTIFY=$iii
/*PASS  wwwwww
/*ROUTE  XEQ DELTA
//DMBATCH EXEC PGM=DMBATCH,PARM=(YYSLYNN)
//STEPLIB DD DISP=SHR,DSN=PCC.NDM32.LINKLIB
//DMNETMAP DD DISP=SHR,DSN=PCC.NDM32.NETMAP
//DMPUBLIB DD DISP=SHR,DSN=PCCSYS.NDM.PROCESS.LIB
//DMSGFIL DD DISP=SHR,DSN=PCC.NDM32.MSG
//DMPRINT DD SYSOUT=*
//*YSUDUMP DD SYSOUT=*
//SYSIN DD *
SIGNON USERID=(iii)
        ESF=YES
SUBMIT PROC=COPY SNODE=NDM.SGA.SDC
        &DSN1=$iii.PCC.WORK.CNTL(IEFBR14)
        &DSN2=SSA.ACF.DC.TEST
        &DISP1=NEW
        &DISP2=CATLG
        &COMPRESS=
SEL PROC WHERE (QUEUE=A) TABLE
SIGNOFF
/*
```

Where:

iii	=	North System registered initials
aaa	=	North System agency code
pppuu	=	North System project code
wwwwww	=	North System password
dataset1 name	=	data set name at the originating data center
dataset2 name	=	data set name at the receiving data center

6.9 TAPE INFORMATION AND UTILITIES

The following sections present tape information for the OS/390 North System. Refer to the Appendix for Titan tape information.

For the current Titan tape information, refer to:

<http://silk.nih.gov/silk/titan>

and follow the links to "Titan system page" and then "tapes."

Also, see

<http://silk.nih.gov/tapes>

and the *Titan User's Guide*.

6.9.1 Tape Management, Security, and Storage

The OS/390 North System uses IBM 3480 magnetic cartridges as the primary storage media for tape oriented data. Only foreign tape processing allows the creation of the round style magnetic reel tape. The reel tape must be purchased following processing.

A tape library is provided for storage of North System-controlled tape. Users are responsible for their own tape data and should backup all essential data sets that are not easily reconstructed.

Tapes using TMS will automatically fall under the protection of TSS for internal security. For additional information, see Section 6.9.1.3.

6.9.1.1 Tape Management System TMS/CA1 Features

The North System uses TMS/CA1, a software system, to maintain accountability of tapes in the tape library. The system, upon initial creation of a cartridge tape, records the tape serial number and the data set name and characteristics in an internal catalog. This catalog permits the NIH Computer Center to:

- verify that the proper cartridge tape is mounted
- produce comprehensive inventories of magnetic cartridge tape allocations by registered agency and individual user

Data set Expiration Dates and TMS

Users should become familiar with the TMS procedure for establishing and maintaining expiration dates of tape data sets. Knowledge of, and adherence to this procedure will prevent inadvertent or premature release of tape data sets. Important points to be considered are:

- If users do not specify an expiration date or retention period on the DD statement when they create or recreate a data set, the system assumes a retention default value of five calendar days. After the fifth day, the tape is automatically released to the "scratch pool" for reuse and is no longer assigned to the creator of the data set.
- The expiration date is specified on the DD statement through either the EXPDT or RETPD subparameters of the LABEL parameter. EXPDT is a 7-digit Julian date and RETPD is from one-to-three digits indicating the number of days to retain the tape from the date of creation.
- The expiration date can be extended or altered by other methods. See Section 6.9.1.6 for details.
- Multi-data set tapes are retained until all expiration dates on the tape have expired.
- Processing a data set for input does not affect its expiration date. However, output processing affects expiration dates in the following manner:
 - The expiration date is reestablished whenever a tape is used for output. The retention period must be specified on the DD statement or the default value of five calendar days will be invoked.
 - If a tape is "OPENED" for output and is not successfully "CLOSED" by the processing program (i.e., it is closed by abnormal termination processing), the retention period for the tape will be five calendar days regardless of what may have been specified on the DD statement. This is based on the assumption that any data set closed by abnormal termination processing is going to be recreated later in a successful run. However, the five-day period allows the user time to take action to keep the data if it has some value. This applies both to creation of new data sets and to recreation of existing data sets.
 - The DISP=MOD parameter of the DD statement, used to append data to an existing data set, does not change the expiration date.

The following examples illustrate a data set with expiration date and retention period specification. In the next example, the cartridge tape will be retained until the 115th day of 2001.

Example 161

```
//DD1 DD UNIT=CTAPE,DSN=AAA.III.MYTAPE1,DISP=(,KEEP),
// LABEL=EXPDT=2001/115
```

In the next example, the cartridge tape will be retained for 130 days from the date of creation.

Example 162

```
//DD2 DD UNIT=CTAPE,DSN=AAA.III.MYTAPE2,DISP=(,KEEP),  
// LABEL=RETPD=130
```

In the following example, since no expiration date is specified, by default the cartridge tape will be released to the scratch pool for reuse after five calendar days.

Example 163

```
//DD3 DD UNIT=CTAPE,DSN=AAA.III.MYTAPE3,DISP=(,KEEP)
```

Tape Security Features

The TMS data set security feature provides user protection of private tape data sets. For additional information on tape security, see Section 6.9.1.3.

6.9.1.2 TMS Error Conditions

Various system completion codes and informative messages can result during TMS processing. The more serious abnormal situations result in the job step being abnormally terminated by TMS with a system completion code of the form nEC where n can be from 1 to 6. See Section 7.2.1 for these completion codes.

Another abnormal situation occurs when a scratch tape is requested for output and TMS does not accept the mounted tape as a scratch. TMS issues a "NOT A SCRATCH" message with an accompanying ERROR code and requests another tape. If the situation resulted from improperly coded DD statements, the job will be terminated. See Section 7.2.1 for this message and associated codes.

6.9.1.3 Tape Security System (TSS)

The North System provides an in-house tape security system, Tape Security System (TSS), which automatically protects tape data sets created at the NIH Computer Center from access by other users. Like TMS, the TSS protects data sets on a volume basis; therefore, all files on a multi-file tape are protected. Under TSS a tape volume can have either one or two owners. An owner's registered initials appear in positions five through seven of the data set name (DSN) or as the first three characters of the creating jobname as contained in the TMS record for the tape data set. In most cases the creating jobname initials and those in the DSN will be the same but they need not be. Owners have automatic access to the volume. All others must appear on the access list in order to have access to the tape. Only a tape owner can modify a North System access list. Adherence to the data set naming conventions for tapes is crucial to this system. Failure to use the proper DSN format will result in an abnormal termination of the job. The acceptable format for tape data set names other than print or plot tapes is:

aaa.iii.....

Where aaa is the agency code which the user is registered; and iii is the registered initials of the user or of another user in the same agency code.

All tapes will be protected by TSS. Access to a tape data set volume is allowed when:

- The requester's initials match the iii portion of the DSN on the tape (as described above aaa.iii)
- The requester's initials are the first three characters of the creating jobname as contained in the TMS record for the tape
- The requester's initials on the access list contained in the TMS record for the tape and has authority (read or write) for the operation in progress
- The requester's agency code is on the access list with the proper authority

Failure to meet one of the above criteria will cause an abnormal termination (ABEND 9xx) to occur.

A number of users create tapes that are subsequently used by others both within and outside of their own agency account. TSS allows other users access to those tape data sets after permission has been granted. The owner of a tape data set grants access to other individuals or to all users within an agency account by placing his registered initials on the access list for the tape data set. Multi-volume data sets are automatically updated whenever any one volume of the set is updated.

There are two ways to access TSS to update the access list of a tape. From the ISPF Primary Option Menu, use P17, or from the READY mode in TSO, enter %TPERMIT and the CLIST prompts for parameters. All the necessary information can also be supplied when the CLIST is invoked as illustrated below.

Example 164

```
%TPERMIT VOLUME(nnnnnn) ID(iii) ACCESS(a)
```

Where: nnnnnn = 6 digit volume serial number
 iii = 3-character user initials or agency code.
 a = R for READ
 W for WRITE
 D for DELETE

Another CLIST, %TLIST, exists to list the contents of a tape's access list. In a similar manner, it is invoked by simply entering %TLIST and answering the prompt or entering %TLIST and answering the prompt or entering %TLIST VOLUME(nnnnnn).

Access to data sets can be granted in batch mode using several techniques. The following examples illustrate the JCL used to grant data set access to others with the first example demonstrating how to grant access to an individual user.

Example 165

```
//iiiPCC JOB (aaa,pppuu),'your name',CLASS=E
/*PASS $$$
//STEP1 EXEC BATCHTSO
%TPERMIT DSN=(aaa.iii.data) ID(iid) ACCESS(a)
/*
```

Where:

iii =	Registered initials for data set owner
iid =	Registered initials to have access
aaa =	Agency account code for owner
ppuu =	Project code for owner
a =	R for READ
	W for WRITE
	D for DELETE

When the data set is not cataloged, the %TPERMIT and the VOLUME=nnnnnn parameter must be substituted for the DSN=(aaa.iii.data) parameter where nnnnnn = reel number. The following example demonstrates how to grant access to all users within an account.

Example 166

```
//iii#TAPE JOB (aaa,xxxxxx)
.
.
//tapeddn DD DSN=aaa.iii.data,UNIT=CTAPE ...
```

Where:

iii =	Registered initials for data set owner
aaa =	Agency account code for owner
# =	Indicates to grant access to all registered users within the owner's account.

The pound sign (#) as the fourth character of the JOB statement automatically grants access to all registered users within the individual user's own account.

TSS and Tape Access Lists

Under TSS anyone (other than the creator) who needs access to a tape has to be on the access list for that tape. Several things—including TSS itself—will help.

- TSO option P.17 is used for this purpose. Individual IDs or group IDs can be placed on the access list.

- TSS interfaces with RACF and the concept of "special users" within a group applies for tape access, just as it does for DASD data sets. See the *RACF Coordinator's Manual* for information on "special users" within a group.
- In TSS, if the fourth character of the creating jobname is a "#" (pound sign), access to any tape created by the job will automatically be given to anyone in the creating organization. Doing this has the same effect as placing the group ID on the access list of any tape created by the job.

6.9.1.4 Data Encryption, Subroutine Method

[NOT SUPPORTED ON TITAN]

There are two subroutines for data encryption: PCCNCODS, used to encode, and PCCDCODS, used to decode a data string with a given alphanumeric key. The subroutines require four input arguments and have an optional fifth argument:

Argument 1	The address of the data to be encoded or decoded.
Argument 2	The address of a full word containing the number of bytes of data to encode or decode.
Argument 3	The address of the alphanumeric encode or decode key in character format.
Argument 4	The address of a full word containing the length of the key. The maximum key length is 255 characters.
Argument 5	The address of a workarea of at least 600 bytes. If this argument is used, ENCODE or DECODE will use the address as a pointer to a working storage area rather than obtain a new area. This feature is provided to reduce system overhead.

The following user abnormal termination codes will be issued when error conditions exist:

2221	invalid number of arguments input
2222	invalid data address input
2223	invalid data length input
2224	invalid key input
2225	invalid key length input
2226	invalid workarea address input

The following example shows statements from a FORTRAN program that encode an 80-byte area with a key of USA7 and provide a working storage area for PCCNCODS.

Example 167

```
DIMENSION DATA(20),WORK(150)
DATA IDL/80/,KEY/4HUSA7/,KEYL/4/

CALL PCCNCODS(DATA,IDL,KEY,KEYL,WORK)
```

The following example shows statements from a COBOL program that decode an 80-byte area with a key of 293571.

Example 168

```
WORKING-STORAGE SECTION.
01 ARG1 PIC X(80).
01 ARG2 PIC S9(8) VALUE=+80 USAGE COMP.
01 ARG3 PIC X(6) VALUE='293571'.
01 ARG4 PIC S9(8) VALUE=+6 USAGE COMP.

PROCEDURE DIVISION.

CALL 'PCCDCODS' USING ARG1,ARG2,ARG3,ARG4.
```

The following example shows statements from a PL/I program that decode an eight-byte area with a key of US1976 and provide a working storage area for PCCDCODS. Since the encryption software uses standard parameter list format, the PL/I Compiler must be instructed to generate the correct list for the call. One such method is to provide the COBOL entry option.

Example 169

```
DCL PCCDCODS ENTRY OPTIONS(COBOL);
DCL DATA CHAR(8);
DCL DATAL FIXED BIN(31,0) INIT(8);
DCL KEY CHAR(6);
DCL WORK CHAR(600) INIT((600)')ALIGNED;

KEY='US1976';
CALL PCCDCODS(DATA,DATAL,KEY,KEYL,WORK);
```

6.9.1.5 Tape Standards

The standard recording mode for NIH Computer Center library cartridge tapes is 18 track, 38000 bpi. Specifications for the label parameters and options may be found in Section 5.5. A list of the parameters required for tape processing is included in Section 5.5.2. All tape data set names must conform to standards as specified in "DSN= (Data set Name) Parameter" in Section 5.5.2. All tapes which are not under the control of TMS must be processed as foreign tapes.

Recreating Tape Data Sets

When recreating a data set on a specific cartridge tape volume, the user must observe the following rules:

- The same data set name must be used. Only when the data set name on the DD statement matches the data set name on the cartridge tape label will the expiration date be ignored and TMS allow the file to be overwritten.
- DISP=OLD must be specified. (DISP=NEW is used exclusively for requesting a scratch volume to create a new data set or adding a new data set to an existing cartridge tape volume).

If both of the above conditions are not met in recreating a data set on a specific volume, a "scratch" volume will be used, and the only indication that this has occurred is in the JES2 Job Log at the start of the job output. For a multiple data set tape, the only data set that can be recreated is the last one.

Bypass Label Processing (BLP)

The use of bypass label processing (BLP), a subparameter of the LABEL=, is not allowed for the North System. It is required that users processing foreign tapes ensure that the tape has been created as non-labeled or with IBM standard labels. Also ensure that the supplier provides the data set name, density, record format, record size, and block size.

If any difficulties are encountered in processing the tape, the user should then contact TASC for further assistance.

6.9.1.6 Tape Inventory Management

The NIH Computer Center allows users to release tapes early, extend expiration dates, or designate tapes for long-term, off-site storage.

OS/390 North System users can run tape listing reports from the Web using SILK (Secure Internet-LinKed) Web technology. Tape reports can be run for an entire agency code or an individual userid.

To run the tape listing report, go to:

<http://silkad.nih.gov/taperpt>

or select "Tape Inventory" from the North System's Web page.

Enter your agency code, and userid if desired, and click on the submit button. The Tape Inventory Report can be viewed immediately on the Web by clicking on the "view report" button after submitting the report. A report can also be routed to a mainframe printer for paper output by specifying "local" or remote printer in the entry box. You can also issue tape permit commands that allow another user to access a specified tape.

The inventory listings contain the following information:

- volume serial number (VOLSER)
- data set name (DSN)
- date created
- date expires
- date renewal (for extending the expiration date)
- date last used
- times used
- creating JOBNAME (JOBNAME that created the tape data set)
- volume sequence number (VOLSEQ)
- tape density
- off-site storage indicator

Each tape scheduled to expire within 60 days of the listing date is highlighted with a "greater than" (>) symbol on the inventory listing.

Users should examine the listing and take the appropriate action for each tape listed:

- To immediately release a tape, circle the volume number.
- To extend an expiration date, indicate the new expiration date (month/day/year) in the "Date Renewal" column.

Users should sign and return the annotated inventory within ten calendar days of the date of the listing to:

National Institutes of Health
Center for Information Technology
Output Distribution Services
Room 1000
12 South Drive
Bethesda, Maryland 20892

Tape Inventory Reminders

There are two methods of receiving reminders for tape expirations:

- The ISPF function P.16 allows account coordinators to review expiration dates for specific tapes by user or by reel number. Extending the expiration date of a tape or deleting tapes from the library can also be accomplished through the P.16 SPF panel.

When working with a list of tapes under ISPF (P.16), consult the options available at the top of the screen. Use the date 99365 to mean permanent when updating individual tapes via option 4 of this facility.

- The listserv list, NORTH-TAPE-L, is available for account coordinators who want to be reminded that their tape inventory needs to be checked for expiring tapes. Those who sign up for this listserv will receive monthly e-mail reminders to check their tape inventory. For information on the listserv facility, go to:

<http://list.nih.gov>

6.9.1.7 Procedure for Purchasing Tape

To create magnetic reel tapes to transfer out, use the following procedures:

- Reel tapes must be user owned or purchased from the NIH Computer Center in advance of processing.
- Jobs creating such tapes must be processed through a North System Foreign Tape Procedure i.e., EXPDT=98000 in the Label parameter and a job class of either "B" or "D."
- UNIT parameter should be **one** of the following options:
 - UNIT=FRGN3480 for 3480 BPI tapes
 - UNIT=FRGN6250 for 6250 BPI tapes
 - UNIT=FOREIGN for 800 or 1600 BPI tapes
- Volume serial number of the tape must be explicitly requested in the VOLSER=.

Users wanting to create 3480 cartridge output to transfer out of the NIH Computer Center may create the output as any standard data set. To do this, create the cartridge with the standard data set naming convention and purchase the tape through Production Control when it is needed.

To generate a tape where the label must conform to some other naming convention, processing the job must be coordinated in advance with Production Control. The tape must be purchased, identified, and processed as a foreign tape.

Removal of Tapes from the Tape Library

To request removal (purchase) of a tape from the tape library, users must complete a tape removal form. Tapes may be removed any time during operating hours.

Removal of a tape from the library results in the following:

- All references to the tape are deleted from the Tape Management System TMS/CA1.

-
- The requester is assessed a fixed charge for each tape removed to cover the cost of replacement.
 - The deleted tape volume serial number is assigned to a new tape that becomes part of the TMS "scratch" tape pool.
 - If at a future time the user wishes to process the removed tape at the NIH Computer Center again, it must be processed as a foreign tape. (See Section 6.9.1.8.)

Removal of Tape

In order to remove tapes, users must obtain property passes from their administrative officer prior to the close of the business day. CIT will not issue property passes for tapes.

6.9.1.8 Foreign Tape Processing

A foreign tape is defined as a tape not under control of the Tape Management System TMS/CA1. The NIH Computer Center will process 9-track foreign tapes recorded at densities of 6250, 1600, 800 bpi., and cartridge tapes of 38000 bpi. There is no 7-track capability at the NIH Computer Center.

The following procedures must be followed to process foreign tapes:

- A foreign tape must be registered at the NIH Computer Center on each day that it is to be processed. Each reel tape must be secured with a tape band. The band must have the volume serial number written on it. An external adhesive label (supplied by the NIH Computer Center) must be attached to the reel and provide the following information:
 - volume serial number (same as written on the tape band)
 - programmer name
 - agency identifier code
 - office mailing address
 - telephone number
 - tape density (38000, 6250, 1600, or 800 bpi)
- All foreign cartridge and reel tapes must be processed using job classes B or D (see Section 5.2.1).
- When allocating a tape drive for mounting a foreign tape, the user must specify one of the generic names for foreign tapes that are provided in the UNIT= parameter of the DD statement: "UNIT=FOREIGN" for 1600 or 800 bpi reel tapes, "UNIT=FRGN6250" for 6250 bpi reel tapes, and "UNIT=FRGN3480" for 38000 bpi cartridge tapes.
- The DD statement allocating a tape drive for a foreign tape must also include the parameter LABEL=EXPDT=98000.

- There are cataloged procedures available for copying or creating foreign tapes. They are discussed below.
- After processing, foreign tapes are returned to Production Control by 7:00 a.m. of the day following registration. If the tape requires additional processing, it must be reregistered by the user in person or by telephone. The maximum amount of time that a foreign tape may stay at the NIH Computer Center is five days; after this period, the tape is returned to the user.

Foreign Tape Copy Procedures

These cataloged procedures are provided to facilitate the processing of data tapes that are not North System library tapes. These procedures meet most of the requirements set forth above in Section 6.9.1.8.

FRGNCOPY will copy a foreign reel tape to a North System cartridge tape. FRGNOUT will copy a North System library cartridge tape to a user's foreign reel tape for removal of the data from the NIH Computer Center. FRGNOUTC will create a 38000 bpi copy of a North System library cartridge tape. CLASS=B or CLASS=D must be specified in the JOB statement.

The following example shows how to copy a standard labeled 1600/800 bpi foreign tape, serial number nnnnnn to a North System library cartridge tape and specify a retention period of 30 days.

Example 170

```
//IIINAME  JOB  . . . ,CLASS=B
//STEPNAME EXEC FRGNCOPY,DSNIN='FOREIGN.LABELED.TAPE',
//  VOLSER=nnnnnn,
//  DSNOUT='AAA.III.PCCTAPE',DAYS=30
```

The following example shows how to copy a foreign 6250 bpi, non-labeled tape to a North System library tape. This tape will be retained for the default period of five days.

Example 171

```
//IIINAME  JOB  . . . ,CLASS=B
//STEPONE  EXEC FRGNCOPY,
//  VOLSER=nnnnnn,LABEL=NL,UNIT=FRGN6250,
//  FM=FB,REC=80,BLK=3200,DSNOUT='AAA.III.PCCTAPE'
```

Symbolic Parameters:

BLK	<i>Required for unlabeled input.</i> - Specifies the block size for blocked input. (Same as DCB BLKSIZE parameter.)
DAYS	<i>Optional</i> - Specify a retention period for the output tape. If not used, a retention period of 5 days will be assigned by default.

DSNIN	<i>Required for labeled input.</i> - Specifies the input data set name.
DSNOUT	<i>Required</i> - Specifies a North System standard name for the data set copied to North System library cartridge tape. (See Section 5.5.2.)
FM	<i>Required for unlabeled input.</i> - Specifies the record format. (Same as the DCB RECFM parameter.)
LABEL	<i>Required for unlabeled input.</i> - Must specify LABEL=NL.
REC	<i>Required for unlabeled input.</i> - Specifies the logical record length of the input. (Same as the DCB LRECL parameter.)
UNIT	<i>Required for 6250 bpi input.</i> - Must specify UNIT=FRGN6250 for reel tape. Must specify UNIT=FRGN3480 for cartridge.
VOLSER	<i>Required</i> - Must specify either the external identifier on unlabeled foreign tape or the internal volume serial number of foreign tapes that are OS standard labeled.

The following example will use the OS standard labeled 1600 bpi foreign tape, serial number 026715 for output, and create a copy of the library cartridge tape 060750.

Example 172

```
//IIINAME  JOB  . . . ,CLASS=B
//STEPNAME EXEC FRGNOUT,
//  DSNIN='AAA.III.PCCTAPE',INSERT=060750,
//  OUTSER=026715,DSNOUT='FOREIGN.LABELED.OUTPUT',DEN=3
```

Symbolic Parameters:

DEN	<i>Required to create an 800 bpi or 1600 bpi output.</i> - Must specify DEN=2 for 800 bpi, DEN=3 for 1600 bpi and, DEN=4 for 6250 bpi output.
DSNIN	<i>Required</i> - Specifies the name of the North System data set to be copied.
DSNOUT	<i>Required</i> - Specifies the data set name of the output when an OS standard label foreign tape copy is being created.
INSERT	<i>Required</i> - Specifies the volume serial number of the North System library tape containing the data set to be copied.
LABEL	<i>Optional</i> - Required for unlabeled output. - Unlabeled output must specify LABEL=NL.

OUTSER	<i>Required</i> - Must specify either the external identifier on unlabeled foreign tape or the internal volume serial number of foreign tapes that are OS standard labeled.
UNIT	<i>FOREIGN</i> - Must specify to create an 800 bpi reel tape. If UNIT is not specified, the default will be a 6250 bpi tape.
INUNIT	<i>Optional</i> - UNIT=TAPE must be specified if the input is a tape reel; the default is cartridge tape (CTAPE).

The following example will use the foreign cartridge tape 010666, and create a 38000 bpi copy of the North System library cartridge tape 056777.

Example 173

```
//IIINAME  JOB  . . . ,CLASS=B
//STEPNAME EXEC FRGNOUTC,DSNIN='AAA.III.PCCTAPE' ,
//  INSER=056777,OUTSER=010666
```

Symbolic Parameters:

DSNIN	<i>Required</i> - Specifies the name of the North System data set to be copied.
DSNOUT	<i>Required</i> - Specifies the data set name of the output when an OS standard label foreign tape copy is being created.
INSER	<i>Required</i> - Specifies the volume serial number of the North System library tape containing the data set to be copied.
LABEL	<i>Optional</i> - Required for unlabeled output. - Unlabeled output must specify LABEL=NL.
OUTSER	<i>Required</i> - Must specify either the external identifier on unlabeled foreign tape or the internal volume serial number of foreign tapes that are OS standard labeled.
INUNIT	<i>Optional</i> - UNIT=TAPE must be specified if the input is a tape reel; the default is cartridge tape (CTAPE).

Foreign Reel Tape/Cartridge Processing

Reel drives for foreign tape processing on the OS/390 North System are only available at the NIH Computer Center's Parklawn site. Foreign 3490 cartridge drives are available for use by the North System on the NIH campus. Foreign 3480 cartridge drives are available at both locations. For the convenience of users, arrangements have been made to permit tapes to be checked in at either site—and then delivered to the other site by courier as necessary.

When processing foreign tapes/cartridges—those not controlled by TMS, the following check-in procedures apply:

Type of Tape	Check-in At	Process At
Reel Tapes	NIH Computer Center or Parklawn	Parklawn only
3480 Cartridges	NIH Computer Center or Parklawn	Parklawn and NIH
3490 Cartridges	NIH Computer Center or Parklawn	NIH only

Access to the Parklawn Building services is at output distribution in Room 2B70.

A courier service operates twice daily between the NIH Computer Center and the Parklawn Building for users wishing to have the tapes/cartridges delivered to either place. Refer to the *NIH Computer Center User's Guide* for the schedule.

The UNIT parameters for foreign tape/cartridge processing are as follows:

FRGN3490	3490 Cartridges processed at NIH Computer Center *
FRGN3480	3480 Cartridges processed at Parklawn
FRGNDCRT	3480 Cartridges processed at NIH Computer Center *
FOREIGN	9-TRACK 1600/800 bpi tape
FRGN6250	9-TRACK 6250/1600 bpi tape

* New unit parameters added to accommodate processing foreign tapes at the NIH Computer Center.

If you have any questions concerning these procedures, please call TASC.

6.10 SANITIZE FOR ERASING DATA

[NOT SUPPORTED ON TITAN]

The SANITIZE utility is a batch program which may be used to erase tape or disk data sets when it is necessary to ensure that data is completely destroyed. The following should be taken into consideration when using this utility:

- Tape reels are erased from the first record after the header to the tape indicator mark at the end of the reel.
- Tape cartridges are overwritten with a random pattern from the first record after the header to the tape indicator at the end of the cartridge.
- Disk data sets are opened for sequential output and will be overwritten with a random pattern from the beginning to the end-of-file mark. Partitioned data sets, therefore, must be erased one member at a time.

- Multi-volume tape files can be processed, but each volume must be specified on a separate DD statement.
- A VSAM data area cannot be erased with this utility; the IBM Access Method Services (AMS) utility must be used.
- A PARM field on the EXEC statement must be provided if multiple files are to be overwritten in a single execution.

The following examples show the JCL format for the overwrite utility.

Example 174

```
//STEPNAME EXEC PGM=SANITIZE,PARM=t
//SYSPRINT DD SYSOUT=A
//FILEn DD ...
"
"
/*
```

NOTES:

- PARM=t, where t is the number of FILEn DD statements, must be included if more than one file is being processed.
- A FILEn DD statement must be supplied to define each input, where "n" is a unique number for each file.
- DCB information is not required.
- Data sets are not scratched or uncataloged unless requested through the JCL (example: DISP=(OLD,DELETE)).

In the following example, a multi-volume cartridge tape data set with private data is being returned to the scratch pool. The file consists of three cartridge tapes. The data set is not cataloged. The EXEC statement PARM field specifies that three files are to be overwritten. The FILE1, FILE2, and FILE3 DD statements each specify one volume of the multi-volume data set.

Example 175

```
//STEPNAME EXEC PGM=SANITIZE,PARM=3
//SYSPRINT DD SYSOUT=A
//FILE1 DD DSN=AAA.IIIMASTR,
// DISP=OLD,VOL=SER=nnnnnn,UNIT=CTAPE
//FILE2 DD DSN=AAA.IIIMASTR,DISP=OLD,VOL=SER=nnnnnn,
// UNIT=AFF=FILE1
//FILE3 DD DSN=AAA.IIIMASTR,DISP=OLD,
// VOL=SER=nnnnnn,UNIT=AFF=FILE1
/*
```

In the following example, a temporary cartridge tape file is created in STEPSTRT, passed for further processing, then overwritten and deleted in STEPFINI. The EXEC PGM=SANITIZE statement has no PARM field, which indicates that only one file is to be overwritten. The FILE1 DD statement specifies the tape created in STEPSTRT.

Example 176

```
//STEPSTRT EXEC IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT2 DD UNIT=CTAPE,DSN=*&TEMP,DISP=( ,PASS)
  (Other processing steps)
//STEPSFINI EXEC PGM=SANITIZE
//SYSPRINT DD SYSOUT=A
//FILE1 DD DSN=*&TEMP,DISP=(OLD,DELETE)
```

6.11 VISION:UILDER (MARK IV) CATALOGED PROCEDURES

There are several cataloged procedures available to facilitate access to the VISION:Builder (MARK IV) software and to assist users in the maintenance of their libraries. See the following list of cataloged procedures and their functions:

MARKUTIL	Library maintenance.
MARKIVD	Dictionary definition run.
MARKIV	Processing run.
MARKDUMP	Dumps a VISION:Builder (MARK IV) library.
MARKREST	Restores a VISION:Builder (MARK IV) library.
MARKCON	Compresses a VISION:Builder (MARK IV) library.

The examples below show the use of the VISION:Builder (MARK IV) cataloged procedures and describe the symbolic parameters provided. It is recommended that MARKUTIL be used for all library maintenance, since the MARKINIT, MARKDUMP, MARKREST and MARKCON utilities will not be supplied in future releases.

MARKUTIL is a service program that is used for establishing and maintaining M4LIBs. It is able to perform all of the following functions: initialization, dump, restore, condense, copy and merge. MARKUTIL functions are specified in the UC statement. The power of MARKUTIL is that multiple UC statements may be included in a MARKUTIL job step (e.g., initializing a library, and then copying selected items into it from another library.) The following example is one-step job that both initializes a new library and condenses an existing library.

Example 177

```
//STEPNAME EXEC PGM=MARKUTIL
//M4LIST DD SYSOUT=A
//M4LIB DD DSN=markiv.init.library,
// DISP=(NEW,CATLG),UNIT=PUB,
// SPACE=(TRK,(n,n),RLSE)
//M4LIB2 DD DSN=markiv.condense.library,DISP=OLD
//M4WORK DD DSN=markiv.work.file,DISP=(NEW,DELETE),
// UNIT=SYSDA,
// SPACE=(TRK,(n,n),RLSE)
//M4INPUT DD *
UCINIT
UCCOND
/*
```

The following example executes a procedure that is a one-step job that allows a user to catalog file definitions, transaction file definitions, transaction definitions, and tables.

Example 178

```
//STEPNAME EXEC MARKIVD,LIB='AAA.M4LIB'
```

The following example is a three-step job that will process data, sort data, and generate report.

Example 179

```
//STEPNAME EXEC MARKIVD,LIB='AAA.M4LIB'
```

The following example is a one-step job that dumps a cataloged VISION:Builder (MARK IV) library from disk to cartridge tape.

Example 180

```
//STEPNAME EXEC MARKDUMP,LIB='AAA.M4LIB',
// TAPEDSN='AAA.IIIXXXXX',RETPD=nnn
```

The following example is a one-step job that restores and compresses a previously dumped VISION:Builder (MARK IV) library from cartridge tape to disk.

Example 181

```
//STEPNAME EXEC MARKREST,LIB='AAA.M4LIB',
// TAPEDSN='AAA.IIIXXXXX',TAPEVOL=nnnnnn
```

7 NORTH - MESSAGES AND CODES

The following section describes messages and codes for the North System.

7.1 COMMON ABENDS

The information in Section 4.1.1 is valid for both the North and South Systems. Additional information for the North System follows:

ABEND Code	Description
S013	Conflicting or unsupported parameters in DCB—member name specified in DD not found; no directory allocation subparameter in DD.
Possible Causes	<ul style="list-style-type: none">• DCB parameters conflict, no BLKSIZE or not a valid multiple of LRECL• missing or misplaced SYSIN DD card• partitioned data set member not found• MERGE - SORTIN01 does not contain the maximum blocksize• JCL - tried to create a PDS without allocating directory blocks
S0C1	Operation exception—program attempted to execute an operation code that is not valid on this CPU.
Possible Causes	<ul style="list-style-type: none">• subscript error – overwrote existing code• tried to read a file that was not opened• misspelled DDNAME• error in parameters passed to subroutines• missing DD card• recording mode was wrong or density was incorrect• bad load module, possible bad object deck
S0C4	Protection exception—a virtual address could not be translated into a real address.

Possible Causes	<ul style="list-style-type: none"> • tried to execute code outside of the program • bad branch • subscript error • tried to move data beyond the end of a buffer, or getmain area; overlaid instruction(s) • invalid address was referenced due to subscript error or bad parameters • the key of an instruction or an operand in storage does not match the protection key in the PSW (interrupt code 4) • the virtual segment or page was never allocated (interrupt code 10 or 11, respectively) • the page was paged out and the routine requesting the code was disabled for I/O interrupts (interrupt code 11) • COBOL errors <ul style="list-style-type: none"> • in group move, receiving wrongly-defined record variable length • tried moving variable length record larger than target field maximum • tried to read or write a file which was not open • misuse of DD dummy statement • tried to call within COBOL's internal F SORT input/output procedure • used with SORT - tried to go back in the output procedure
-----------------	---

0C7 Data exception—decimal data is incorrect or improperly overlapped or not validly initialized.

Subscript error—expecting to find packed data, but packed data is not present; could be binary or EBCDIC; or table was not initialized.

Failure to initialize a variable or array.

Possible Causes	<ul style="list-style-type: none"> • subscript error, referenced beyond table • COBOL errors <ul style="list-style-type: none"> • working storage not initialized • bad data, check data for errors • period missing after imperative statements within at end clause • binary field in an arithmetic operation is not large enough to accept result
222	The operator or the job entry subsystem cancelled the job. Check the job log listing for more information. If there is no apparent explanation, contact operations before resubmitting.
Possible Causes	<ul style="list-style-type: none"> • failed to free a dataset needed by your job • multiple jobs in execution after the same tape line or card estimate exceeded • JCL error caused mount request for invalid or non-existing volume • requested devices not currently available (job may be re-queued)
322	Job or step time exceeded the specified limit.
Possible Causes	<ul style="list-style-type: none"> • program is in a loop • insufficient time parameter on JOB or EXEC card
B37	At end of volume, the volume must be demounted, but the system is unable to dismount the volume.
Possible Causes	<ul style="list-style-type: none"> • the data set on DASD output already had 16 extents, but required more space • secondary space was too small • no more space was available on the volume • the volume table of contents was full
D37	Partitioned dataset is full.

-
- Possible Causes
- 100% full (check: ISPF 3.2)
 - directory blocks all used up (check: ISPF 3.2, or SPACE DIR DSN DIR)
 - alternative: compress it (if 100% full) by doing one of the following:
 - (1) %compress dataset
 - (2) use ISPF/PDF 3.1, option 1 to display information on directory blocks
 - (3) batch—IIEBCOPY utility
 - alternative: recreate it, with more space and/or dir blocks
 - (1) %RECREATE, or
 - (2) RENAME it, ALLOC a new larger dataset, COPY old to new.

7.2 TMS COMPLETION CODES AND MESSAGES

The completion codes and messages generated by the Tape Management System are provided below.

7.2.1 TMS SYSTEM ABEND CODES

During normal processing, certain error conditions may be noted. When an error condition exists, a system generated ABEND is invoked:

1xx - Invalid Data Set Name

Explanation: The data set name in the JCL does not agree with the data set name in the Tape Management Catalog (TMC) for the volume involved.

User Response: Verify that the data set name in the JCL is correct and/or the correct volume was requested.

2xx - Invalid Secondary Data Set

Explanation: The user is trying to open the second or higher data set on a tape volume.

User Response: If the tape is being opened for input processing, verify that the JCL contains the correct file sequence number and the correct volume was requested. If the tape is being opened for output processing, the message IECTMS3 contains a reject code that describes the error. (See Section 7.3.)

3xx - Invalid TMC Record

Explanation: An invalid volume serial number or DSNB pointer has been found.

User Response: Make sure the volume serial number is entered correctly. If it is correct, contact TASC.

4xx - Invalid Multi-Volume Data Set

Explanation: The multi-volume file being opened for input is not the same volume that the data set was created on.

User Response: Verify that the correct volume serial number was specified in the JCL.

6xx - Invalid Password

Explanation: An attempt was made to open a password protected data set and the user did not supply the correct password.

User Response: Supply the correct password.

7xx - Dismount of a DISP=MOD tape

Explanation: A MOD tape has been rejected.

User Response: See the IECTMS3 message in Section 7.3 for the reject code that describes the error.

FEC - Invalid SVC (IBM ABEND)

Explanation: The error occurred during execution of an invalid supervisor call instruction.

User Response: Refer to the IBM messages and code documentation available through the CIT publication ordering service.

1yy - IPL Required

Explanation The TMC has been extended or moved.

User Response: Contact TASC.

2yy - All Data Set Name Blocks Are Allocated

Explanation: There are no secondary data set name blocks (DSNBs) that can be allocated for this request.

User Response: Contact TASC.

3yy - Invalid DSNB Record

Explanation: An invalid DSNB or control record has been found.

User Response: Contact TASC.

4yy - Invalid Parameter List

Explanation: The parameter list passed to SVC ED contains an invalid field.

User Response: Verify that no user modification is causing the ABEND.

7yy - Backup is needed immediately

Explanation: The audit data set is full.

User Response: Contact TASC.

Dyy - I/O Error has Occurred

Explanation: An I/O error has been detected on the audit data set.

User Response: Contact TASC.

Eyy - I/O Error Has Occurred

Explanation: An I/O error has been detected on the TMC data set.

User Response: Contact TASC.

Fyy - Invalid SVC (IBM ABEND)

Explanation: The error occurred during the execution of an invalid supervisor call instruction.

User Response: Refer to the IBM documentation for messages and codes.

7.3 JES2 MESSAGE LOG - TMS MESSAGES AND REPLIES

nn IECTMS1 DDD, , ENTER VSN

Explanation: An NL or BLP tape is being opened for input or output.

User Response: Reply with the correct volume serial number. If a specific mount request was made, only one matching reply is required. If a non-specific request was made two successive, matching replies are required.

nn IECTMS2 DDD,VVVVVV,VERIFY TAPE FROM OUTSIDE LIBRARY

Explanation: A duplicate volume serial number in the TMC range is being opened for output and the JCL specifies LABEL=EXPDT=98000.

User Response: Reply 'M' to unload the tape so the correct tape can be mounted. Reply 'U' to verify that the tape is indeed from outside the user's library and is not under TMS control.

IECTMS3 DDD,VVVVVV IS NOT SCRATCH (xx)

Explanation: IECTMS3 A tape mounted as 'scratch' is unacceptable as 'scratch' to TMS.

User Response: Mount another tape as 'scratch'. The tape is unloaded. A tape may be determined not to be 'scratch' for a number of reasons. The condition code (xx) in the IECTMS3 message indicates the reason for rejecting the 'scratch' mount. The condition codes are defined below:

- 04 = Operator replied 'M' to a '98000' request (see IECTMS2)
- 08 = VSN not in the TMC and EXPDT = '98000'
- 12 = Operator replied 'M' to a '98000' EXPDT check (see IECTMS4)
- 16 = Operator did not reply a specific NL VSN (see IECTMS1)
- 20 = TMC EXPDT = 99365 (permanent hold)
- 24 = Volume marked OUT-OF-AREA
- 28 = TMC Data Set Name = HDR1 DSN
- 32 = Volume has not expired (not processed by TMSCLEAN)
- 36 = Same volume remounted for EOVS output request
- 40 = Volume part of valid multi-volume chain

44 = Specific VSN required (see IECTMS7)
48 = Volume has DSNB's (trying to create 1st file)
52 = Trying to create file 'N' and 'N-1' does not exist
56 = Trying to create file 'N' and 'N+1' already exists
60 = Trying to create file 'N' but 'N' already exists
64 = Trying to recreate a data set with DISP=NEW
68 = Label check occurred at open on file sequence greater than one
72 = Label check occurred at open on password protected volume
76 = Trying to recreate tape with RETPD=0

nn IECTMS4 DDD,VVVVVV IS UNEXPIRED

Explanation: The expiration date in the HDR1 record of an output tape not controlled by TMS (i.e., XPDT=98000) is greater than the current date.

User Response: Reply 'M' to dismount the tape. Reply 'U' to override and write on the tape.

IECTMS6 DDD,VVVVVV IS APPROVED FOR LABEL CHANGE

Explanation: A TMS controlled tape has been approved for label change. The console operator can re-label the volume in response to OS messages.

User Response: None.

IECTMS7 DDD,VVVVVV, SPECIFIC REQUEST

Explanation: A specific request for volume VVVVVV is being made at EOVS.

User Response: Mount the requested tape on drive DDD.0

7.4 OS/390 NORTH SYSTEM RACF MESSAGES

02I NEW PASSWORD CANNOT EQUAL CURRENT PASSWORD

Description: An attempt was made to change a password when the new password was the same as the old password.

User Response: Make the new password different from the old password.

PCC0001I NOTICE - PASSWORD HAS BEEN ACTIVE FOR 90 DAYS

Description: The password has not been changed during the past 90 days.

User Response: The recommended procedure is to change your password at least every 90 days.

PCCX0003I --WARNING-- PASSWORD EXPIRES IN nn DAY(S)

Description: The password is about to expire in "nn" days. This message will be issued on every job beginning 5 days prior to the expiration date.

User Response: Change the password before the expiration date.

PCCX0004I --WARNING-- PASSWORD EXPIRED FOR nnn DAY(S)

Description: The password has expired and a change is required. The current password will be accepted for 5 days after expiration.

User Response: Change the password immediately.

PCCX0005I JOB REJECTED -- PASSWORD EXPIRED FOR nnn DAY(S)

PCCX0005I LOGON REJECTED -- PASSWORD EXPIRED FOR nnn DAY(S)

Description: The password expired more than 5 days ago and must be changed.

User Response: Change the password immediately.

PCC8002I INVALID NEW PASSWORD

Description: An invalid word was entered as a new password. The new password cannot be the word "LOGON" or cannot be equal to the userid.

User Response: Provide a valid new password.

RFX204I DATASET LEVEL SET ON NON-STANDARD DSNAME

Description: The level indicator in the data set profile was set with a project access authority but the data set did not have the project code in the proper position in the data set name. This message is accompanied by the RACF message ICH408I.

User Response: Contact the owner of the data set to request a change to the access authorization level or to rename the data set so that the project code is in the proper position.

RFX205I CONFLICT OF USER GROUP AND ORGANIZATION CODE IN DSN

Description: The organization code in the dsname was not the same as the group name of the user who attempted access and the user was not specifically permitted to use the data set via the access list or the UACC authority. This message is accompanied by the RACF message ICH408I.

User Response: Contact the owner of the data set to request access authorization.

RFX206I USER DOES NOT HAVE JOB CARD PROJECT VERIFICATION

Description: The project level indicator in the data set profile was set, but the user is not required to run batch jobs using specific project codes. This message is accompanied by the RACF message ICH408I.

User Response: Contact the owner of the data set to request access authorization or contact the RACF coordinator to require project verification.

RFX207I NO INSTALLATION DATA IN USER PROFILE

Description: The project level indicator in the data set profile was set, but the user is not required to run batch jobs using specific project codes. This message is accompanied by the RACF message ICH408I.

User Response: Contact the owner of the data set to request access authorization or contact the RACF coordinator to require project verification.

RFX208I USER NOT AUTHORIZED TO USE PROJECT

Description: The project level indicator in the data set profile was set, but the user is not authorized to run batch jobs using the project code

indicated in the dsname. This message is accompanied by the RACF message ICH408I.

User Response: Contact the owner of the data set to request access authorization or contact the RACF coordinator to include the project code in the user profile.

RFX209I --WARNING-- UNDEFINED DATASET LEVEL (XX)

Description: The level indicator in the data set profile was set to an undefined level indicated by "XX". This message is accompanied by the message RFX299I.

User Response: Contact the owner of the data set.

RFX210I --NOTICE-- SPECIAL PRIVILEGE ACCESS AUTHORITY FOR GROUP

Description: The user's profile permits access to all organization data sets for the requested level of access. This message is accompanied by the message RFX299I.

User Response: No response required.

RFX211I SPECIAL PRIVILEGE INCORRECTLY SPECIFIED IN USER PROFILE

Description: The "special privilege" indicator was present in the installation data section of the user profile, but the organization code was not the same as the current connected group.

User Response: Contact the RACF coordinator to correct the user profile.

RFX299I datasetname VOL(volser)

Description: This message is accompanied by either a warning or error message.

User Response: Refer to the warning or error message that accompanies this message to determine user response.

RFX301I --WARNING-- USER (XXX) GRANTED ACCESS (ALTER/CONTROL)

Description: This message is accompanied by messages RFX308I and

RFX399I. A data set has been created where the current user described in message RFX308I does not match the userid fields in the dsname. The user in the dsname is automatically granted ALTER/CONTROL authority for the data set described in message RFX308I and RFX399I.

User Response: Check the dsname for possible spelling error.

RFX302I DSNAME ERROR - INCORRECT ORGANIZATION CODE IN DSNAME

Description: An attempt has been made to create a data set where the organization code in the dsname does not match the current group of the users. The data set is described in messages RFX308I and RFX399I.

User Response: If the organization code in the dsname is correct, contact the organization to request UPDATE access authority.

RFX303I --NOTICE-- RACF PROTECTED, ACCESS VIA BASE INDEX
VOL(GDGVOL)

Description: A new generation to a Generation Data Set Group has been created, but a RACF data set profile has not been defined to RACF. Subsequent access to that generation will be through the base index that should have been previously defined to RACF.

User Response: Verify that the base index has been defined to RACF on volume GDGVOL using NOSET.

RFX308I USER(iii) GROUP(aaa)

Description: This message is accompanied by message RFX399I and either a warning or error message.

User Response: Refer to the warning or error message that accompanied this message to determine user response.

RFX399I datasetname VOL(volser)

Description: This message is accompanied by message RFX308I and either a warning or error message.

User Response: Refer to the warning or error message that accompanied this message to determine user response.

7.5 ABR ONLINE MESSAGES FOR DATA SET RETRIEVAL

FDRW70 ELIGIBLE FOR RECALL -- DSN.AAA.III.XXXXXX.DATA

FDRW71 TYPE 'UC' TO UNCATALOG THE DATA SET

FDRW71 TYPE 'YES' TO PERMIT THE RESTORE OR PRESS 'ENTER' TO
BYPASS

Explanation: The automatic recall function is in effect for online processing.

User Response: Press the 'ENTER' key to cancel the recall. Enter 'UC' to
uncatalog the data set or 'YES' to restore.

FDRW76 DATA SET IS SCHEDULED TO BE RESTORED TO VOLUME SERIAL
NUMBER PUBnnn

FDRW77 TYPE NEW VOLUME SERIAL NUMBER AS XXXXXX OR PRESS
'ENTER' TO CONTINUE

Explanation: The recall function permits the data set to be restored to the
original storage volume or to another volume.

User Response: Enter a new storage volume number or press the 'ENTER' key
to restore the data set to its original volume.

FDRW78 RESTORE START COMMAND ISSUED -- YOU WILL BE NOTIFIED
UPON COMPLETION

Explanation: While the data set is being restored, the terminal is available to
perform other tasks.

User Response: Proceed to perform other tasks while waiting for the restore to
complete.

FDRW 311 FOR RESTORED DSN=AAA.III.XXXXXX.DATA ALLOCATED
CATALOGED CN(00)

Explanation: The data set is ready for immediate use.

User Response: None Note: In batch mode, jobs simply pause and wait for the
data set to be restored before continuing to execute.

Note: In batch mode, jobs simply pause and wait for the data set to be restored before
continuing to execute.

8 APPENDIX

- 8.1 - Selected pages from the Titan transition Web site – <http://silk.nih.gov/silk/titan> – including the Titan Transition home page, the Titan and North system summaries of design changes by topic, and detailed topic pages. Most topic pages include two aspects: the general design on Titan, followed by specific changes that former North system users need to make.
Hint: To keep up to date with the latest changes, print out new Web pages and insert them into this manual.
- 8.2 - Selected articles from Titan/South News – <http://datacenter.cit.nih.gov/titannews>
- 8.3 - Additional NJE information for TITAN

8.1 APPENDIX – SELECTED TITAN TRANSITION WEB PAGES

<http://silk.nih.gov/silk/titan>

Transition to Titan

The new MVS standard system is known as "Titan." This system combines the best features of the NIH Computer Center's *North* and *South* systems into a single "standard" environment, providing significant benefits to our customers.

How to Navigate this Site

Users may find it easier beginning with the [Titan system page](#), which provides an overall summary of the design of the new system. There are links from the Titan pages to pages describing changes for specific topics as they relate to North or South users.

This site will continue to change during the transition, so check back on a regular basis.

If you would like to be notified each time this site is updated, join the Standard system [Listserv](#).

** If you would like assistance with moving your application to Titan, CIT staff will be glad to come and meet with your group. Please set up an appointment by contacting TASC at 4-6248 (301-594-6248) or by sending e-mail to TASC@nih.gov.*

July 27, 2001

Titan Specifics Summary

Area Of Change	Summary Of Change	Date Last Updated
USERid	The format of new USERids has changed. Old USERids have been copied to Titan.	March 14, 2002
Accounts	Old accounts are still valid.	March 14, 2002
Data Set Naming Conventions	Existing North data sets are accessible from Titan.	March 14, 2002
Logon	The Telnet and FTP addresses for accessing Titan are new. Also, the Titan logon screen has a new format.	March 14, 2002
Batch Job JCL	In general, most existing JCL should work on Titan.	March 14, 2002
Software	Most North and South software is available on Titan.	August 16, 2001
Tape Data	Access to tape data sets is via established RACF permissions.	March 14, 2002
Disk Data	The North system and Titan share all disk data.	March 14, 2002
RACF Security	RACF profiles protect all disk and tape data on Titan.	May 1, 2001
Printed Output	Header and trailer pages look different. Any change to routing of output to printers is described here.	March 14, 2002
Databases	Special consideration must be given for users of database systems.	May 1, 2001

Data Transfer	Information about changes needed for CONNECT:Direct users, etc.	March 14, 2002
-------------------------------	---	----------------

Billing and Rates	New billing system for Titan.	March 14, 2002
-----------------------------------	-------------------------------	----------------

March 14, 2002

Transition to Titan

North system Summary of Changes


North System Specifics

The North migration began on January 2, 2001 and will complete July 31, 2001. Please also see the system specifics for [Titan](#) for details about what this system will look like.

Summary of Changes

North system users may observe changes in a number of areas. Several general areas where changes will occur are listed below. Please refer to the appropriate area of concern to you.

Area Of Change	Summary Of Change	Date Last Updated
USERid	The format of new USERids has changed, but old USERids have been copied to Titan.	March 14, 2002
Accounts	Minor changes in agency/organization codes. Old agency codes are still valid.	March 14, 2002
Data Set Naming Conventions	Existing North data sets are accessible from Titan.	March 14, 2002
Logon Changes	The Telnet and FTP addresses for accessing Titan are new. Also, the Titan logon screen has a new format.	May 1, 2001
Batch Job JCL Changes	Some users may need to change their JOB cards for jobs being submitted. In general, most existing JCL should work on Titan.	August 16, 2001

Software	Most, but not all North software is available on Titan.	March 14, 2002
Tape Changes	North system tapes can be accessed from Titan. Access to tape data sets will be via established RACF permissions.	July 27, 2001
Disk Data Sets	North and Titan systems share all disk data.	May 1, 2001
RACF Security	Establishing data set protection can only be done from the North system until the North-to-Titan conversion is complete.	May 1, 2001
Printed Output	Header and trailer pages look different. Any change to routing of output to printers is described here.	July 27, 2001
Databases	Special consideration must be given for users of database systems, such as ADABAS and Model 204. Read this for important information.	May 1, 2001
 Data Transfer	Information about changes needed for CONNECT:Direct users, etc.	July 27, 2001
Billing and Rates	New billing system for Titan.	May 1, 2001

March 14, 2002

NIH Computer Center

Titan Transition - USERids

USERids, Accounts and Data Set Naming Conventions

USERids

Ids are used when accessing system services such as batch jobs, interactive and database systems, and RACF. One of the goals for simplifying the Titan environment was to permit a single id to be used (e.g., for the TSO and RACF ids). When moving to Titan, users will encounter some minor changes in this area.

A USERid identifies a particular registered user of a system. On Titan, the USERid, TSOid and RACFid are identical. The USERid:

- 1) validates individuals signing on to the system (like the TSOid)
- 2) is the high-level qualifier for user-owned data sets
- 3) identifies users who are permitted access to data (like the RACFid)

Although the North and South naming conventions differ greatly, the USERid scheme designed for Titan will accommodate both the current North and South USERids with no need to rename disk data sets.

While the migration progresses, all USERids will be restricted to the current North and South formats. Once the North and South systems have migrated, Titan USERids may then be from 3 to 7 characters long, with the first character an alphabetic letter or a \$. Each USERid will be associated with **one and only one** account. **Note:** Any user application that captures the RACFid must be able to accept a RACFid that is from 2 to 8 characters in length.

User-owned data sets begin with the USERid -- with the form **userid.name** (for example, johndoe.dataname) -- or with the account (for example, aaaa.dataname).

[Information for North system customers.](#)

[Information for South system customers.](#)

Accounts

On Titan, accounts will not change from those used on the North and South systems. All existing North system agency codes (accounts) and South system accounts will be copied to Titan as accounts.

All accounts on Titan will be defined as RACF groups, and the USERids assigned to an account will be members of that RACF group.

Data Set Naming Conventions

On Titan, data set names (disk or tape) must begin with either a USERid (userid.dataname) or a Titan account (aaa.dataname or aaaa.dataname). This complies with the RACF requirement that data set names begin with a USERid or a RACF group, because Titan accounts are defined as RACF groups.

Data set names that begin with an account/RACF group (aaa.dataname or aaaa.nnn) are ideal for production work. Because they do not belong to an individual, they do not need to be renamed if the USERid is reassigned.

[Information for North system customers.](#)

[Information for South system customers.](#)

Examples

New Titan users could observe the following:

USERid or RACFid or TSOid	johndoe
Account or RACF Group	aaa <i>or</i> aaaa
Dsname Convention	johndoe.dataname <i>or</i> aaa.dataname <i>or</i> aaaa.dataname

March 14, 2002

NIH Computer Center

Titan Transition - North USERids

USERids and Dataset Naming Conventions

USERids

Currently, North system USERids are of the form **iii**, associated with a single 3 character agency code/account. North system USERids (iii) and RACFids (iii) match, but the TSOid (\$iii) differs. North user-owned disk data sets begin with the TSOid -- with the form **\$userid.name** (for example, \$iii.dataname).

All North system USERids have been copied to Titan and **contain a leading \$**. North system USERid **iii** has become USERid **\$iii** on Titan.

On Titan, you will continue to logon to TSO with the string **\$iii** and you should not have to rename any data sets.

Applications that capture the RACFid: If your application is expecting to handle 3 characters, it may need to be modified. Any user application that captures the RACFid must be able to accept a RACFid that is from 2 to 8 characters in length.

To ease migration, all North TSOids (\$iii) were brought over to become the USERid/RACFid/TSOid on Titan. South system USERids propagated to Titan will be 7 characters in length. Once the North and South systems are completely migrated to Titan, USERids/RACFids/TSOids can then be from 3 to 7 characters long, with the first character an alphabetic letter or a \$.

Registering New Users: Account sponsors/coordinators who need to register a new user should go to [Web Sponsor for Titan](#) and select the link *Add New Customer to an Account*.

Dataset Naming Conventions

Currently, North system data sets begin either with the TSOid (\$iii.dataname) or the agency code, followed by the USERid (aaa.iii.dataname). North users will not need to rename any of their disk or tape data.

The North convention of beginning the data set name with the agency followed by the USERid (aaa.iii.dataname) is accepted on Titan but not enforced. The only Titan requirement for data set names is that they begin with either a USERid (\$iii.dataname) or account (aaa.dataname).

Examples

In the table that follows, North initials are shown as **iii** and the agency/account as **aaa**. When you move from the North system (left columns), you will see the following changes on Titan (right columns):

On North		On Titan	
USERid	iii	USERid	\$iii
Agency	aaa	Account	aaa
RACFid	iii	RACFid	\$iii
TSOid	\$iii	TSOid	\$iii
RACF Group	aaa	RACF Group	aaa
Dsname	\$iii.dataname <i>or</i> aaa.iii.dataname	Dsname	\$iii.dataname <i>or</i> aaa.iii.dataname

March 14, 2002

NIH Computer Center

Titan Transition - Logging On

Logon Changes

New Telnet and FTP Addresses

Because Titan co-exists with the South system during the transition period, Titan's telnet addresses must differ from the South system's telnet (**tn3270.cu.nih.gov**) and FTP (**ftp.cu.nih.gov**) addresses.

Titan Access is via the addresses:

tn3270.titan.nih.gov for Telnet,
ftp.titan.nih.gov for FTP access,

3270 Logon Screen

On Titan the 3270-type access logon screen differs from the South system screen. When you get to the Application Selection screen, you will need to type NIHTSO to access the Titan system.

Screen Scrapers

Some users' applications make use of screen scrapers -- that is, software that reads and extracts information from the output screen. Many of these applications will need to be changed to accommodate the Titan TCP/IP address (tn3270.titan.nih.gov) and the Titan logon screen format.

.

March 14, 2002

NIH Computer Center

Titan Transition - Batch & JCL

Batch Jobs

One of our goals in merging the North and South systems was to make the transition as easy and transparent as possible. A new system facility, ThruPut Manager, greatly aids in achieving this goal.

ThruPut Manager -- a software product developed by MVS Solutions Inc. -- can convert the JCL submitted by a North or South user to run on Titan. It recognizes the format of an incoming North or South job statement, converts the JCL, and directs output to the appropriate printers. **In general, existing JCL should work on Titan.**

ThruPut Manager can also, as its name suggests, control the flow of jobs in Titan. It can recognize when a job needs tapes and schedule the job accordingly. For example, it can identify jobs that require tape mounts and delay a job from starting if the job requires more tape drives than are currently available. As Titan develops and settles in, other features of ThruPut Manager will likely be used to increase efficiency.

JCL Changes

We have tried to minimize required changes to JCL. We recommend that you first try running your existing JCL as is on Titan to see whether any changes will be necessary to migrate your batch work to Titan from the North or South systems.

While most existing JCL should work on Titan, some users may need to change their job statements to reflect the new RACFid in the USER= field. If a password is needed it can be included on the job statement as ",PASSWORD=..." The system automatically fills the USER= field with the TSOid of the submitting user. Therefore, jobs submitted from a TSO session (or Model 204 session) do not need to have USER= or password coded, unless the job is being submitted under a different USERid than the one used to log on.

Job Statements

For ease of transition, TITAN will automatically recognize North and South system job statements and JES2 /* statements and execute the jobs in the appropriate Titan job class and at the appropriate time. Hence for any JCL

streams you currently may have, no job statement or JES2 /* statement changes are necessary.

Titan also gives you the option of using a new, simpler Titan job statement format. If at any time you need to change your job statement, we recommend that you replace it with the new Titan format. The Titan job statement follows the conventions documented in the IBM JCL Manuals. An example of a Titan job statement is:

```
//jobname JOB (),pgmrname, TIME=time,LINES=nnn,...
```

All parameters in italics are optional. Thus the simplest valid job statement on Titan is:

```
//jobname JOB
```

Please refer to the *OS/390 MVS JCL Reference* manual for details of the various fields.

Note: On Titan, a jobname does not need to be a USERid -- it can be any alphanumeric string up to 8 characters. Jobs are then fetched in IOF (Interactive Output Facility) based on the USER= field of the job statement rather than the jobname. If USER= is not explicitly specified, the USERid of the TSO submitter is assumed on the job. If the job is submitted by another job, the USERid will propagate from the first job to the second job.

The "USER=" parameter (for example, USER=\$iii) must be added to any job statement that is submitted from an RJE workstation or another NJE node.

Box Numbers

South box numbers will remain the same on Titan. Titan output boxes located at the Parklawn site have the same numbers as the North boxes -- with an added prefix of "P."

Users who run jobs at other NJE nodes and then route the job to print on Titan must add a /***JOBPARM ROOM=bbbb** statement after their JOB statement, where "**bbbb**" is the box number where the output is to be placed. If you want the output placed in one of the boxes in the Parklawn building, place the letter "**P**" in front of the box number. Without this /***JOBPARM ROOM=bbbb** statement, your output may not print or may go to the wrong box.

Job Classes

Titan has two job classes.

Class L -- Long Job Class -- with maximum CPU time of 59940 seconds. This is the default job class if you do not code a TIME= on your job statement to indicate that your job needs 100 CPU seconds or less for processing.

Class X -- Express Job Class -- with a maximum of 100 CPU seconds. Jobs of this class will be searched first when JES2 is selecting a job to begin execution. For new JCL streams that follow the simpler Titan job statement

standards, class X can not be directly specified. Any job statement with TIME=(1,40) or less will automatically run in the Express execution class.

On a Titan job statement, you will need to include the proper ThruPut Manager statements to request overnight or weekend processing. To request overnight processing, the following control statement should be added:

```
/*+JBS BIND OVERNITE
```

Similarly, to request weekend processing add this control statement to your JCL stream:

```
/*+JBS BIND WEEKEND
```

ThruPut Manager automatically converts existing North and South system job classes into the appropriate Titan job class.

Routes to PCCJES2 node

South system users and users from other systems (such as CDC) who previously routed jobs to run or print on the North system will need to change the destination JES2 nodename in their JCL.

Central Printing:

from **PCCJES2** to **NIHJES2**

VPS or Remote Printing:

from **PCCJES2**,*printrname* to
NIHJES2,*printrname*

from **PCCNJEDV**,*printrname* to
NIHJES2,*printrname*

Execution:

from: /*XMIT **PCCJES2** to: /*XMIT
NIHJES2

Sending Email from a Batch Job

The following JCL can be used to send email in batch on Titan:

```
=====
// EXEC SENDMAIL
//MAILIN DD *
from: senderid@mail.nih.gov
to: recipient@mail.nih.gov
subject: subject of email
```

```
body of email
more body
end of email
```

THE MAILIN dd card can be any dataset, with LRECL less than or equal to 240, which contains the email lines.

- The 'FROM:' record is required and should specify one valid email address. Any delivery error messages will be sent to this address. Since Titan currently has no facility for receiving email, the email address should be on some other email system where you already receive and read your email.
- The 'TO:' record is required and should contain one or more email addresses separated by blanks, commas, or semicolons. In addition, multiple 'TO:' records can be specified.
- The 'SUBJECT:' record is not required, but should normally be specified.

Additional email header records:

cc: recipient@mail.nih.gov
reply-to: reply@mail.nih.gov

- The 'CC:' record can contain one or more email addresses and there can be more than one 'CC:' record.
- The 'REPLY-TO:' record, if used, should contain one email address to receive replies to the email.

March 14, 2002

NIH Computer Center

Titan Transition - North Batch & JCL

Batch Jobs Job Classes

ThruPut Manager automatically converts existing North system job classes into the appropriate Titan job class. North system job classes E, B, and A become Titan job class X (express), and North system job classes C, D, H, F, R, and S become Titan class L (long). The North system F, R and S classes that request overnight or weekend processing will automatically have ThruPut Manager syntax statements added so that execution will commence at the appropriate times.

August 16, 2001

NIH Computer Center

Titan Transition -- Software

Software Available

In order to assist users in making as smooth a transition as possible to Titan, CIT has copied most existing cataloged procedures and system software from the North and South systems to Titan.

Cataloged Procedures

Most cataloged procedures will be available on Titan, so it is expected that most users' jobs will run on Titan with little or no change. The exceptions pertain to software identified below that is not moving to Titan.

Compilers

The NIH Computer Center supports compilers for three high-level languages -
- COBOL, FORTRAN and PL/I.

On Titan, only the current versions of these compilers will be supported. This will affect users who have not recompiled the source programs for their applications for a number of years. Users of some of the older versions of these compilers will need to recompile and re-link edit when they move to Titan. The versions of these compilers in use on Titan are:

COBOL For MVS And VM, Version 1.2.2
IBM PL/I For MVS And VM, Version 1.1.1
VS FORTRAN, Version 2.6

Some Software Not Available on Titan

Some software currently used on the North and South systems will not be moved to Titan. For example, the older compilers used on the North system will be replaced by current versions. Additionally, some other software products will not be moved.

August 16, 2001

NIH Computer Center

Titan Transition - North Software

Software Available

In order to assist users in making as smooth a transition as possible to Titan, CIT has copied most existing cataloged procedures and system software from the North and South systems to Titan.

Cataloged Procedures

Because some system software will be changed or upgraded, some procedures have been modified to work correctly on Titan. For example, the COBOL procedures will be changed to use most current version of the COBOL compiler, and all SORT procs have been modified to use DFSORT in place of Syncsort.

Compilers

The NIH Computer Center supports compilers for three high-level languages -
- COBOL, FORTRAN and PL/I.

On Titan, only the current versions of these compilers will be supported. This will affect users who have not recompiled the source programs for their applications for a number of years. Users of some of the older versions of these compilers will need to recompile and re-link edit when they move to Titan. The versions of these compilers in use on Titan are:

COBOL For MVS And VM, Version 1.2.2
IBM PL/I For MVS And VM, Version 1.1.1
VS FORTRAN, Version 2.6

Please note that these are all newer versions than currently exist on the North system. The versions of these compilers on the North system are:

COBOL For MVS And VM, Version 1.2.1
PL/I Optimizing Compiler, V1.4
VS FORTRAN, V2.3

Warning - Different Versions Can Cause Problems during the Transition to Titan

Although a program that is compiled and link edited on the North system will generally work on Titan, the reverse is not true -- a program compiled and link edited on Titan will very likely fail to run on the North system. Care must be taken when compiling application programs during this transition. For users

who create libraries containing load modules (called load libraries or loadlibs), we recommend that you create separate loadlibs on the North and Titan systems. After you have completely moved to Titan, back up and delete the old load libraries. If your application always compiles a source program and then executes it, there should be no problems.

Some Software Not Available on Titan

Some software currently used on the North and South systems will not be moved to Titan. For example, the older compilers used on the North system will be replaced by current versions. Additionally, some other software products will not be moved.

If the removal of any of the following facilities will cause problems with your application, please call TASC.

Syncsort -- replaced by IBM SORT on Titan *

Pccvtoc -- to be removed

Pccncode and Pccdcodc -- to be removed

Sanitize -- to be removed

Oracle -- to be removed

Dynamic Model II (DYNAMO II) -- to be removed

General Purpose Simulation System (GPSS) -- to be removed

* This change will not require any action by the user. The products are syntactically compatible, and the system procs point to the new product.

March 14, 2002

NIH Computer Center

Titan Transition - Tapes

Tape Changes

The tape management system that Titan uses is CA-1, from Computer Associates. User access to the tape database for tape management (displaying information, extending expiration dates, cataloging and uncataloging, or scratching tape data sets) is through ISPF panels. Batch and Web access may be added in the future.

Duration of Tape "Ownership"

On Titan, you must define the expiration date for tape data at the time you create it. In your JCL code a specific expiration date or retention period for the newly created tape data set or specify that the tape should be kept either permanently or until the data set is uncataloged. **JCL that creates a new tape data set and does NOT include expiration information (EXPDT= or RETPD=) will FAIL.**

Unitnames

On Titan, the unitnames TAPE and CTAPE refer to the virtual 3490 standard tapes in the NIH tape library (see the section on the Virtual Storage Manager). The following unitnames refer to incoming foreign/special tapes or tapes to be written outside of the VSM for removal from NIH. These are for tapes not under control of the tape management system:

FRGN6250 -- 9-track 6250/1600 bpi tape unit (the default output density is 6250 bpi)

FRGN3480 -- 3480 cartridge tape unit.

FRGN3490 -- 3490 cartridge tape unit.

Tape Data Set Names

On Titan, tape data set names must begin with a valid USERid or RACF account group (account). (For more details, see the section on [Data Set Naming Conventions](#).)

Although it is not required, **we strongly recommend that South system users catalog tape data sets when possible.** Cataloged tape data sets will migrate to Titan with very few problems.

Tape Security

Tape data set security on Titan is handled by RACF permissions on a data-set rather than volume basis. Data set protection will depend on the RACF profiles in place and will apply to any data set, regardless of whether it is on tape or disk. Users may need to modify their RACF profiles to provide the same level of access to tape data as they now have.

Virtual Storage Manager

On Titan, most customer tape data resides in the Virtual Storage Manager (VSM). The VSM -- a comprehensive tape storage system consisting of tape silos, disk buffers, new tape technology, and tape management software -- improves performance and reduces human intervention in storing, retrieving, and mounting tapes. All VSM tapes are numbered above 500000.

The NIH tape library will contain only 3490 cartridge tapes. The only 9-track and 3480 tapes kept at NIH will be in a scratch pool. If a user creates a 9-track or 3480 tape from this pool, the tape will automatically be purchased and removed from the NIH Computer Center. All 9-track and 3480 tapes will be treated as foreign/special tapes.

Foreign / Special Tapes

On Titan, there is a check-in and a check-out process for tracking special tapes. Special tapes can be checked in for **up to** a week, to be returned to output boxes the **Monday** following check-in. To make the most efficient use of tape-drive resources, all foreign tapes are mounted at the Bethesda campus, although Parklawn people can continue to check them in at the Parklawn building.

March 14, 2002

NIH Computer Center

Titan Transition - North Tapes

Tape Changes

Since CA-1 is currently used by the North system, North system users should experience few differences. North system tape data is currently available to both the North system and to Titan (see the sections on [Tape Security](#) and the [Virtual Storage Manager](#)).

Duration of Tape "Ownership"

JCL that creates a new tape data set and does NOT include expiration information (EXPDT= or RETPD=) will FAIL.

North system users are familiar with coding expiration dates for tapes. On Titan, you will have to code this information rather than relying on the current North system 5-day default.

Unitnames

These unitnames are currently used on the North system. No JCL changes should be required.

Tape Data Set Names

The current North data set naming conventions are acceptable on Titan, so no change to current names is required.

Tape Security

Since North system users have protected tapes on a volume basis, this will be a change for North users. The RACF generic profiles in use for disk data will be used to protect tape data as well. Option P.17 (TPERMIT) in ISPF -- for protecting tapes on a volume basis -- will not exist on Titan.

If an agency wishes to continue the use of the North-system tape naming conventions of **aaa.iii.dataname**, the RACF coordinator should create RACF profiles of the form **aaa.iii.**** and should permit the **\$iii** USERid ALTER access to that profile. Optionally, the RACF coordinator can set the owner for the **aaa.iii.**** RACF profile as **\$iii**. This will allow the **\$iii** user to use RACF commands to permit other users access to data sets on tape. Unless these RACF permissions are given, the generic profiles currently in use for DASD will be used on Titan and some tape jobs may fail.

Virtual Storage Manager

North system tapes were moved to the VSM as of October 30, 2000. As tapes were copied to the VSM, they were assigned new serial numbers above 500000.

Cataloged tape data sets were automatically recataloged as they were copied to the VSM. If your tape data sets were cataloged, no JCL changes were needed to reflect the new volume serial numbers so long as the data sets were accessed through the catalog -- that is, via the data set name rather than by the volume serial number.

If the tape data set was not cataloged, users needed to modify their JCL to point to the new volume serial number. North users can display information for non-cataloged tape data sets through ISPF Option L.4 (the old P.16) or by executing the TAPEINV CLIST.

Since the VSM is shared between the North and Titan systems, now that all North tapes have been moved to the VSM, the North system tape migration to Titan is complete.

Foreign / Special Tapes **UPDATED**

To make the most efficient use of tape-drive resources, all foreign tapes are mounted at the Bethesda campus, although Parklawn customers can continue to check them in at the Parklawn building.

There are two daily courier runs between Bethesda and Rockville -- morning (8:00 A.M.) and early afternoon (around 2:00 P.M.) -- to pick up and deliver foreign tapes and deliver printed output.

Tape pickup: Parklawn customers can either put their tape(s) in their output box or slide them under the window at the CIT Offsite Distribution Center located in the Parklawn Building (room 2B-70). All foreign tapes need to be logged-in on the sheet located by the window. If the tape is in an output box, you should indicate which box it is in.

If the normal courier times (8 A.M. and 2 P.M.) are okay, you don't need to do anything else. The operator who comes at those times to deliver output will check the window and the log and pick up any tapes in the output boxes indicated in the log. If you need a special pick-up of the tape you can use the phone at the window to call the shift supervisor. The phone number and instructions are posted.

Tape return: All foreign tapes checked in during the week will be returned to your output box the Monday morning following tape check-in.

July 27, 2001

NIH Computer Center

Titan Transition - Disk Data

Disks

To ease migration, disk data sets will be shared between the South system and Titan. That is, data created on the South system can be accessed on Titan, and data created on Titan is accessible from the South system. Disk sharing between the two systems should be in place by the end of March, 2002. At that time, the migration of the South system will begin.

March 14, 2002

NIH Computer Center

Titan Transition - North Disk Data

Disks

Disk data created on Titan can be read or written from either the North or Titan systems. However, there are some limitations for data allocated to unitnames other than those already in use on the North system (e.g., files allocated through TSO/ISPF without a unitname). Attempts to delete or rename this data from the North system will usually fail with one of the error messages below:

Deallocation failed for data set '\$iii.dsname'

IEC614I func FAILED-RC rc, DIAGNOSTIC INFORMATION IS...

Once the transition from the North system to Titan is complete, CIT plans to change how disk data is managed -- using System Managed Storage (SMS) and DFHSM. Largely transparent to users, this change will result in more efficient management of disk data and more options for users.

April 25, 2001

NIH Computer Center

Titan Transition - RACF

RACF Changes

The North RACF database was copied to Titan to minimize changes in RACF processing on Titan. Later, the South RACF data base will be propagated to Titan. What this means is that RACF passwords, RACF profiles and RACF permissions from the North (and later the South) are the same on Titan.

RACF Passwords

On Titan, RACF passwords consist of 6 to 8 alphabetic or numeric characters. RACF passwords will expire every 6 months (180 days). In addition, when a RACF password expires, you may not reset it to any of the previous 5 values your password has had.

Access to Data

All disk and tape data on Titan must be protected by a RACF profile.

May 1, 2001

NIH Computer Center

Titan Transition - North RACF

RACF Changes

Making RACF Changes on Titan

On September 1, 2000, a Titan RACFid was created for every North system RACFid, and all existing RACF profiles and groups were copied from the North system to Titan. Since then, North system RACF protection has been synchronized with the Titan RACF facility. Any RACF changes made on or after September 1, 2000 have been automatically propagated to Titan. So if you create a RACF profile for a data set on the North system, the corresponding RACF profile will be created on Titan.

During the North transition, RACF changes on Titan must be made from either the North system or through Web RACF. Because of the complication of propagating RACF permissions between North and Titan, there is no TSO access to RACF facilities on Titan at this time. A Web RACF for Titan is available but currently only permits RACF password changing and JES functions (permitting others to fetch and purge your batch jobs). Changes to RACF profiles and RACF Coordinator functions should be made from the North system Web RACF facility at this time.

RACF Passwords

If you change your password on the North system, it will be set to the same value on Titan. For example, if you change the RACF password for RACFid **iii** on the North system, the password for the associated RACFid **\$iii** on Titan will be changed to the same value at the same time.

RACF Authorities

On the North System, the RACF coordinator can grant a user special data access privileges across an account's data by including an *AAA, +AAA, or #AAA in the Installation Data Field of the user's RACF profile.

- READ authority to all account data sets is granted with *AAA
- UPDATE authority to all account data sets is granted with +AAA
- ALTER authority to all account data sets is granted with #AAA

This control is implemented with exits and those exits will not be carried over to Titan. To implement privileges on Titan, the RACF coordinator can do the following:

- READ authority (*aaa) can be given by PERMITting the user with READ authority to the AAA.** profile (and to more specific profiles that have been created for the account if appropriate)
- UPDATE authority (+aaa) can be given by PERMITting the user with UPDATE authority to the AAA.** profile (and to more specific profiles that have been created for the account if appropriate)
- ALTER authority has two levels.
 - The first level provides access to data. If the user only creates and deletes data sets that match the AAA.** (or more specific profile) then PERMITting the user with ALTER authority to the AAA.** profile is sufficient.
 - The other level provides authority to the RACF profiles themselves. If the user needs to create RACF profiles for AAA data sets, CONNECTing them to the AAA group with CREATE authority will allow them to create and delete profiles. They cannot, however, change or delete AAA profiles that someone else has created.

The RACF coordinator can use the North system WEB RACF facility (at <http://silkad.nih.gov>) to provide READ, ALTER and UPDATE permissions to data using the option "Adduser to access list to data set (PERMIT)". Please contact the Computer Center if you need to provide CREATE authority to an individual. The ability to provide CREATE authority will be added to Web RACF.

May 1, 2001

NIH Computer Center

Titan Transition - Printing

Printed Output

Print Differences

Header and trailer pages of output printed on Titan differ from those pages on the South systems.

All Titan output is printed at the NIH Bethesda site. Output for boxes prefaced by a "P" will be delivered to the Parklawn Building.

Box Numbers

South box numbers will remain the same on Titan. Titan output boxes located at the Parklawn site have the same numbers as the North boxes -- with an added prefix of "P."

Jobs run at other NJE nodes and then routed to print on Titan must include a **/*JOBPARM ROOM=bbbb** statement after the JOB statement, where **"bbbb"** is the box number where the output is to be placed. If you want the output placed in one of the boxes in the Parklawn building, place the letter **"P"** in front of the box number. Without this **/*JOBPARM ROOM=bbbb** statement, your output may not print or may go to the wrong box.

Routes to PCCJES2 node

South system users and users from other systems (such as CDC) who previously routed jobs to print on the North system will need to change the destination JES2 nodename in their JCL.

Central Printing:

from **PCCJES2** to **NIHJES2**

VPS or Remote Printing:

from **PCCJES2.printername** to
NIHJES2.printername

from **PCCNJEDV.printername** to
NIHJES2.printername

The “USER=” parameter (for example, USER=\$iii or USER=aaaaiii) must be added to any job statement that is submitted from an RJE workstation or another NJE node.

VPS Firewall Configuration for Titan

As applications move to Titan, network administrators for LANs with firewalls may need to make changes in their firewall configuration for VPS to continue working. VPS is a network printing service that permits output from the OS/390 systems to print locally on your network-attached printer.

Existing VPS firewall configurations will continue to work on the South system until applications using VPS are moved to Titan. Once an application is moved to Titan, however, firewalls must be reconfigured because Titan has different IP addresses. Administrators will need to add the following rule:

- source IP address 128.231.64.34
- source port number can be between 721-731
- destination port must be 515

We recommend that administrators make the modification now in order to be ready for Titan. Once application owners have moved their applications to Titan, administrators should delete their old firewall rules permitting VPS access from the South system.

CIT will assist administrators in modifying configurations for VPS printing from Titan. To request assistance, please call TASC.

Once applications have moved to Titan, VPS users who are experiencing printing difficulties should contact their system administrator or technical LAN coordinator (TLC) to ensure that firewalls and print servers are configured correctly.

March 14, 2002

NIH Computer Center

Titan Transition - North Printing

Printed Output

Print Differences

All Titan output is printed at the NIH Bethesda site. Output for North customers will be delivered to the Parklawn Building.

Customers in the Parklawn building will continue to use the CIT Offsite Distribution Center at Parklawn (room 2B-70). Printed output will be delivered to the Parklawn building twice daily -- morning (8:00 A.M.) and early afternoon (around 2:00 P.M.).

Output Hold Limits

The Output Hold limit on Titan is 7 calendar days -- this includes weekends. This timeframe was chosen to be a reasonable compromise between the North system's limit of 45 days and the South system's limit of 24 hours.

Box Numbers

Titan output boxes located at the Parklawn site have the same numbers as the North boxes -- with an added prefix of "P" (for Parklawn). All Titan USERids propagated from the North system have a default output box number prefaced by a "P". If you run a batch job on Titan, the output will be routed to your Parklawn box number unless you specify otherwise. If your North box number is 22, Titan output will be directed to box P22.

If you include a /*BOX statement in your JCL, you need to include the "P", since the /*BOX will override any default box number.

When the operators are sorting the output, any output that has a box number that begins with a "P" will be placed in the box corresponding to that number in the Parklawn building. If the box number does **NOT** begin with a "P," then the output will be placed in the box corresponding to that number in Building 12A at NIH. Output printed at NIH -- but destined for a Parklawn box -- will be sent by courier to the Parklawn building.

Note: The output will be placed according to the box number only. For example, if a job contains a /*ROUTE OUTPUT NIHCU statement but the box number begins with a "P," then the output will be placed in the box at Parklawn.

Special forms and Character Sets

In order to clean up obsolete system data sets, programs, libraries, etc. we copied over to Titan's image library only those forms control buffer (FCB) images, universal character set (UCS) images, FORMDEFs, PAGEDEFs, and character sets (CHARS) that we know are still being used. Unfortunately, there is no foolproof way to determine every image library member in use.

While transitioning to Titan, some users' jobs did not initially print (or got an error message) because a necessary print element had not yet been copied. Once the missing items were copied, the jobs printed. If you have any JCL specifying "FCB=", "UCS=", "FORMDEF=", "PAGEDEF=", or "CHARS=" please verify that your job will print correctly on Titan. If it does not print, please call TASC and ask to speak to a Titan consultant.

3835 Printer to Be Discontinued

The IBM 3835 printer on the North system has reached the end of its useful life. Instead of moving to Titan it will be removed from service when the transition from the North system to Titan is complete. The IBM 3835 printer produces a continuous sheet 8 ½ x 11 form that is burst and trimmed -- and is the default printer for jobs that specify BURST=Y or FORMS=x290 in their JCL.

North system output printed on the 3835 printer will print on the cut-sheet IBM 3160 printer on Titan. This switch should be transparent -- the output produced by the 3835 should be identical to that produced by the 3160 printer. An advantage of the 3160 printer is that you can use both sides of the paper in duplex mode. If you encounter any problems with output from the 3160 printer on Titan, please contact TASC.

July 26, 2001

NIH Computer Center

Titan Transition - Databases

Databases

Unlike individual data sets that can be shared between systems (see the section on Disk Data Sets), sharing a database between two systems is not easily done and will not be done during the North and South transition to Titan. CIT will contact owners of each database application and work with them to move their applications to the new system.

Database Applications

Note: It is not unusual for database applications to submit batch jobs. Some users have embedded JCL in DB2, Model 204 or Natural (ADABAS's language) code. Since it may be necessary to make changes in some JCL, it is possible that changes will be necessary in some of these embedded JCL job streams.

April 30, 2001

NIH Computer Center

Titan Transition - North Databases

Databases

ADABAS

ADABAS users will need to sign on to TSO with **\$iii**. Some ADABAS applications that use the TSOid for internal security may need to change. It may be necessary to change database names for some applications. If this is necessary, then the database name in submitted batch jobs will have to be changed. At this time, it is not clear if or when a database name change will be required. Most of these changes will be on an application-by-application basis. We will update this information on the Titan Web page as we learn more during the transition.

Since ADABAS databases cannot be shared across systems, it will be necessary for users to use new databases on Titan for the testing of their applications.

CICS

While not strictly a database, the Customer Information Control System (CICS, a transaction server) is needed to access and manage critical enterprise data. North system users will need to sign on with **\$iii** rather than **iii**. Scripts that access CICS may need to be revised to accommodate this change. Since the CICS data files can be shared between the North and Titan systems, you may use existing test CICS data files for testing your application on Titan. However, since only one CICS region can access the files at one time, the North system test application must be shut down during Titan testing.

Model 204

Except as noted above for applications submitting batch jobs, no specific Model 204 changes are anticipated at this time.

Since Model 204 databases cannot be shared across systems, it will be necessary for users to establish new databases on Titan for the testing of their applications.

April 30, 2001

NIH Computer Center

Titan Transition - Transferring Data

CONNECT:Direct

CONNECT:Direct provides data transfer capabilities between computing systems in a robust manner. Since CONNECT:Direct still exists on the OS/390 South system, a different naming convention is needed to distinguish this facility on Titan.

New Node Names and IDs on Titan Remote users who transmit data to applications via CONNECT:Direct will also be affected when these applications move to Titan. They will need to make changes in their SNODE specification, and have their VTAM administrators make changes to the VTAM definitions and CONNECT:Direct netmap at their site.

The new CONNECT:Direct node name is NIH.STD.NDM. The APPLid is NIH4NDM, and the NETid is NIH.

Users will need a valid Titan USERid and RACF password to use CONNECT:Direct.

Administrators of remote systems should add the above CONNECT:Direct node definitions to their net map files and also add the VTAM definition to their VTAM tables. While these additions should be made as soon as is convenient, changes to the SNODE specification should not be made until the application actually moves to Titan. We do suggest, however, that a test transfer be done when all system changes have been made.

March 14, 2002

NIH Computer Center

Titan Transition - North Data Transfer

CONNECT:Direct

As North users of CONNECT:Direct move their applications to Titan, they will need to make some changes. The following North system CONNECT:Direct changes will be needed:

- `//DMNETMAP DD DISP=SHR,DSN=PCC.NDM32.NETMAP`

becomes on Titan

```
//DMNETMAP DD
DISP=SHR,DSN=NIH.NDM.NETMAP
```

- `//DMPUBLIB DD DISP=SHR,DSN=PCCSYS.NDM.PROCESS.LIB`

becomes on Titan

```
//DMPUBLIB DD
DISP=SHR,DSN=NIH.NDM.PROCESS
```

- `//DMMSGFIL DD DISP=SHR,DSN=PCC.NDM32.MSG`

becomes on Titan

```
//DMMSGFIL DD DISP=SHR,DSN=NIH.NDM.MSG
```

There is no need for a STEPLIB statement.

Note: Unlike the North system, added security on Titan requires a valid USERid and RACF password to use CONNECT:Direct.

July 26, 2001

8.2 APPENDIX – SELECTED TITAN /SOUTH SYSTEM NEWS ARTICLES

<http://datacenter.cit.nih.gov/titannews>

Titan's Web Sponsor Replaces the North System's RACFREG

On October 16, Titan's [Web Sponsor](#) replaced RACFREG, the OS/390 North user registration system.

What Titan's Web Sponsor Can Do

North system account sponsors now have an easy-to-use Web interface for customer registration and account management, similar to what the South system has. Web Sponsor allows account sponsors to:

- acquire userids for new or existing customers
- change customer information
- remove userids
- perform numerous security and other functions for accounts
- display customer and account information

All customer information, userids, and agency information from the North system registry were replicated into the Titan customer registry used by Web Sponsor. While Titan Web Sponsor provides the same functionality as RACFREG, these terms have changed:

North System	Titan
agency codes	accounts
RACF coordinators	primary sponsors
	security coordinators

Using Web Sponsor

When Titan's Web Sponsor is opened on a browser, the account sponsor is prompted for userid and password. Once entered, the userid and password are remembered by the browser and passed to the Web Sponsor application until the browser is closed.

Using Web Sponsor, account sponsors can affect only their own accounts or userids registered to those accounts. Full documentation for Web Sponsor is provided on the "Web Sponsor" Web page—use the "Documentation" link under Miscellaneous.

If you have questions about Web Sponsor, please call TASC (301.594.6248).

December 21, 2001

Data Set Encryption on Titan Being Considered

Several security features are being incorporated into the Titan system—for example, RACF profiles and file transfers using SSL. CIT is considering whether there is also a need to provide users the ability to encrypt files stored on Titan.

If your application has a need for encryption, please submit a service request ticket online or call TASC at (301) 594-6248. A security consultant will contact you to discuss your application's security requirements.

1/28/02

New Message for Batch Jobs that Access Tapes

A naming convention for tape data sets has never been enforced on the South OS/390 system; however the NIH Computer Center has long recommended that customers preface data set names with a valid account/initials pair—preferably the same as the owner of the tape. On Titan *all* data sets must begin with a valid USERid, whether on tape or disk.

To help prepare for the migration to Titan, the following informational messages will be issued on the South OS/390 system when reading or writing a tape data set that does not begin with a valid account and initials. These messages will appear in the JES2 job log beginning January 7 and will NOT cause your job to fail.

```
15.45.16 JOB07028 NIHCTT1: ****Warning**** VOLSER = 200613 OPEN
FOR OUTPUT
15.45.16 JOB07028 NIHCTT2: Tape Dataset name does not start
with a valid AAAAIIII
15.45.16 JOB07028 NIHCTT3: JOBNAME = jobname DDNAME = SYSUT2
15.45.16 JOB07028 NIHCTT4: DSN=FILE1.BAD.NAME2
15.45.16 JOB07028 NIHCTT5: For more information see URL
silk.nih.gov/tapedsn
```

These messages are intended to help you identify tape data set names that will need to be changed before the job can run on Titan. We do NOT recommend renaming (i.e. copying) *existing* tape data sets at this time—we DO recommend naming *new* tape data sets with the account/initials of the tape owner.

On Titan a tape will be "owned" by the account/initials pair used for the first data set on the tape. The transition will be easiest for those users who have valid data set names that are cataloged.

December 21, 2001

8.3 APPENDIX – ADDITIONAL NJE INFORMATION FOR TITAN

The NIH Computer Center, in cooperation with other data processing centers, provides a capability whereby a batch job can execute at one data center and output can be printed at another. This facility is easy to use and does not require significant JCL changes to implement. NJE (Network Job Entry) files can be transferred between the NIH Computer Center and other computer centers using host-to-host file transfer. This type of file transfer is performed using two batch programs—SENDFILE and RCVFILE.

These programs can perform the file transfer under the following conditions:

- The other site (computer center) must be defined to NIH as a JES2 NJE node; i.e., it must be an OS/390 JES2 or JES3 node, a VM RSCS node, or a VSE POWER node.
 - The node name for Titan is NIHJES2
 - The node name for the South System is NIHCU
- The other site must have the NIH Computer Center-written SENDFILE and RCVFILE programs (or equivalent programs for VM or VSE).
- Users transferring a file must be authorized to use both the sending and receiving site.
- Any job that is submitted from another NJE node or RJE (remote) to run on Titan must have:

`,USER=userid` coded on the JOB statement (where "userid" is the Titan RACFid under which the job will run)

and

`,PASSWORD=password` coded on the JOB statement (where password is the Titan password for the RACFid under which the job will run)

- In order to view the job in OUTPUT HOLD on Titan instead of at the originating site, add a `/*ROUTE PRINT NIHJES2` statement.
- Users who run jobs at other NJE nodes and then route the job to print on Titan must add a `/*JOBPARM ROOM=bbbb` statement after their JOB statement where "bbbb" is the box number where the output is to be placed. If you want this output placed in one of the boxes at the Parklawn site, place the letter "P" in front of the box number.

For more information on SENDFILE and RCVFILE and the sites that support this type of file transfer, refer to Section 6.8.2 and Titan documentation.

9 INDEX

- &PUBLIC BACKUPS, 204
- * (N)
 - parameter of DD, 267
- * or DATA parameter (N), 269
- /*ACCESS, 77
- /*AFTER, 73
- /*AFTER (N), 284
- /*ASSIGN, 77
- /*BEFORE, 73
- /*BEFORE (N), 284
- /*BOX (N), 284
- /*CNTL, 73
- /*DATA, 90
- /*DDOUT statement, 84
- /*DISCOUNT, 78
- /*IBM, 90
- /*INSERT, 90
- /*JOBOUT, 82
- /*JOBPARM (N), 285
- /*KEYWORD, 78
- /*MERGE, 90
- /*MESSAGE, 75
- /*MESSAGE statement in error, 253
- /*NETACCT, 81
- /*NETACCT (N), 286, 302
- /*NOPURGE, 88
- /*NOTIFY, 78
- /*NUMBERED, 90
- /*OUTPUT (N), 286
- /*OVERLAY, 90
- /*PASS (N), 288
- /*PASSWORD, 80
- /*PRINT (N), 288
- /*PRIORITY, 79
- /*PRIVATE (N), 289
- /*QUICK, 79
- /*RERUN, 80
- /*ROUTE, 73, 81
- /*ROUTE (N), 289, 303
- /*ROUTE XEQ, 76
- /*ROUTE XEQ LATE, 76, 81
- /*ROUTE XEQ RECOVERY, 213
- /*SIGNOFF (N), 291
- /*SIGNON (N), 291
- /*TITLE, 90
- /*TSO, 90
- /*UNNUMBERED, 90
- /*XMIT, 80, 235
- /*XMIT (N), 291, 302
- ABENDs
 - common, 241
 - critical messages, 241
 - dedicated disk access, 192
 - DISP to handle data sets, 37
 - F82, 250
 - handling on EXEC statement, 21
 - manuals, 243
- ABENDs (N)
 - common, 329
 - TMS, 332
- above the line. *See* also extended address space
- ABR (N), 280, 281
 - CLISTs, 283
 - features, 281
 - ISPF panels, 282
 - messages, 341
 - TSO facilities, 283
 - WYLBUR commands, 283
- access method services (N), 294
- accounting information, 14, 258
 - NIH Data Warehouse, 228
- ACCT (N)
 - parameter of EXEC, 264
- ADABASE (N), 376
- ADDRPSC (N)
 - parameter of EXEC, 264
- ADSERASE procedure, 133
- ADSMAP procedure, 130
- ADSRECOV
 - backup table capacity, 200
 - control statements, 204
 - LIST statement functions, 203
 - optional space parameters, 208
 - parameters
 - other, 209

SET statement functions, 203
 ADSRECOV procedure, 201
 update table, 200
 allocating space, 44
 AM (N)
 parameter of DD, 267
 AMASPZAP
 restriction, 101
 use MEMALTER, 178
 Appendix, 343
 Titan transition Web pages, 344
 archiving (N), 281
 ASCII data
 DCB subparameter needed, 35
 ASCII data on tapes
 COPY restriction, 105
 utility restriction, 101
 Assembler, 225
 assistance, 1, 2
 Automated Backup And Recovery (N).
 See ABR
 automatic batch job submission, 93
 AVGREC DD parameter, 52
 backup (N), 280
 backups. *See also* disks
 dedicated disks, 191
 batch job cancellation (N), 292
 batch job standards. *See* job standards
 batch jobs
 executing WYLBUR commands, 232
 interactive submission, 92
 programs to submit, 226
 recalling data sets, 170
 submission via theWeb, 92
 batch processing
 introduction, 1
 BDAM data sets
 job delays, 92
 billing
 listing last month's bill, 228
 Binder
 functions, 222
 load module block sizes, 224
 SIZE parameter, 222
 BLKSIZE, 31
 block size

load module, 224
 on DD statement, 31
 recommended, 32
 BLP (N), 317
 BookManager (N), 2
 BUFNO DCB parameter, 34
 BURST (N)
 parameter of DD, 267
 canceling jobs (N), 292
 cards, 14
 cards per job
 specifying, 14
 cataloged data sets
 catalog described, 101
 creating, 62
 listing, 130
 mapping, 130
 naming, 63
 naming restrictions, 102
 cataloged procedures, 66, 67, 99. *See*
 utilities
 adding DD statements, 69
 assigning values, 68
 do not copy, 66
 how to use, 67
 modifying EXEC parameters, 68
 nullifying symbolic parameters, 68
 overriding DD statements, 69
 CATDS procedure, 102
 Center for Information Technology. *See*
 CIT
 charging
 /*DISCOUNT to reduce costs, 78
 for copy of a bill, 228
 reducing CPU time, 32
 CHARS (N)
 parameter of DD, 267, 270
 CHKPT (N)
 parameter of DD, 267
 CICS (N), 376
 class (N)
 job class summary, 261
 CLASS (N)
 parameter of JOB, 259
 COBOL, 225
 codes

completion, 15, 95, 170
combining data sets. *See* concatenating data sets. *See* concatenating input
comment statement, 91
common ABENDs (N), 329
COMPARE procedure, 123
comparing data sets, 123
completion codes, 15, 95, 170. *See* error messages
compressing dedicated disks, 193
concatenating data sets, 61
 example, 26, 34
 merging data sets, 122
 sort input, 118
COND (N)
 parameter of EXEC, 264
 parameter of JOB, 259
COND parameter, 15, 21
condense a PDS
 PDSPACK, 181
condition codes, 15, 170
 error message, 95
CONNECT:Direct, 234, 238
CONNECT:Direct (N), 308
contention. *See* data set contention
COPIES (N)
 parameter of DD, 267
COPY procedure, 105
courier service, 324
CPU time
 estimates, 14
critical messages, 241
cross system enqueue, 72, 91
DATA (N)
 parameter of DD, 267
Data Definition statement, 24. *See* DD statement
DATA parameter (N), 269
data set contention
 30 minute limit, 92
 avoiding, 92
data set integrity, 72
data set names
 error message, 247
 restrictions, 40
data sets. *See also* tape data sets

/*CNTL to avoid contention, 72
cataloging, 101
cataloged
 creating and deleting, 62
 comparing, 123
 concatenating input, 118
 copying, 105
 generation, 228
 merging, 117, 122
 naming conventions, 40
 passed, 65
 printing, 109
 scrambling/unscrambling, 124
 sorting, 117
 temporary, 61
 uncataloging, 101
 utilities for tape and disk, 104
data sets (N), 269
Data Town, 228
data transfer between computer centers, 200
Data Warehouse. *See* NIH Data Warehouse
database changes (N), 375
DATACLAS DD parameter, 54
dataencryption (N)
 subroutines, 315
DB2
 /*CNTL statement, 73
DCB (N)
 parameter of DD, 267
DCB DD parameter, 30
DCB subparameters, 31
 BLSIZE, 31
 BUFNO, 34
 DEN, 36
 DSORG, 34
 for tapes, 36
 LRECL, 31
 OPTCD, 34
 RECFM, 31
DD parameters for creating data sets (N), 269
DD statement (N), 265
 ddname field, 266
 operation field, 267

parameters field, 267
 DD statement parameters
 AVGREC, 52
 CHARS, 28
 check list, 25
 COPIES, 28
 DATA, 26
 DATACLAS, 54
 DCB, 30
 DDNAME, 37
 DEST, 28
 DISP, 37
 DLM, 26
 DSNAME, 40
 DUMMY, 42
 for input stream data sets, 26
 for output stream data sets, 28
 forbidden, 25
 LABEL, 43
 LIKE, 53
 SPACE, 44
 SYSOUT, 28
 UNIT, 46
 VOLUME, 49
 DD statement parameters (N)
 * or DATA, 269
 CHARS, 270
 DEST, 270
 DISP, 270
 DSN, 271
 FCB, 272
 FREE, 273
 HOLD, 273
 LABEL, 273
 SPACE, 275
 SYSOUT, 275
 UNIT, 276
 DD statements, 26
 cataloged data sets, 62
 choosing DD names, 25
 for temporary data sets, 61
 HISTORY, 197
 limit per step, 26
 overriding or adding, 69
 passed data sets, 65
 purpose, 24
 STEPLIB, 66
 use, 24
 DDNAME
 DD parameter, 37
 DDNAME (N)
 parameter of DD, 268
 ddname field (N), 266
 DDSCOPY procedure, 190
 DDSGET procedure, 167
 DDSLIST procedure, 189
 DDSSAVE procedure, 166
 hints, 170
 deallocate data set (N), 273
 dedicated disks
 backups, 191
 maintaining, 191
 maintenance guidelines, 191
 technical coordinator, 191
 defining symbolic parameters, 67
 delays
 avoiding for jobs, 91
 density
 tapes, 36
 DEST (N)
 parameter of DD, 268, 270
 device names
 generic, 102
 DFSORT program, 117
 direct access data sets. *See* DDS
 disk data (N), 367
 disk data sets. *See also* DD statements,
 EDIT format, PDS
 ADSRECOV, 201
 allocating space, 44
 copying from tape, 152
 copying to tape, 152
 creating and using, 61
 DDSCOPY, 190
 DDSLIST, 189
 DSRENAME procedure, 170
 DSSCR procedure, 171
 EDIT format explained, 214
 GET procedures, 152
 getting a report, 130
 JCL for using, 64
 MEMADD procedure, 176

MEMALTER, 178
 memrenam procedure, 177
 naming conventions, 40
 PDSALLOC procedure, 174
 PDSCOPY, 182
 PDSDEL procedure, 175
 PDSLIST, 186
 PDSMAP procedure, 180
 PDSPACK procedure, 181
 procedure to rename, 170
 procedures for EDIT format, 214
 procedures for PDSs, 172
 procedures to scratch, 171
 restoring lost, 201
 SAVE procedures, 152
 saving and retrieving, 169
 scratch space limit, 4
 scratching, 171
 scratching multiple data sets, 172
 temporary, 61
 uncataloging, 101
 unmoveable forbidden, 30, 34
 USRPDSC procedure, 184
 USRPDSL procedure, 180
 utilities summary, 150
 utilities to handle, 105, 149
 disk data sets (N)
 ABR management tool, 280
 disk storage (N), 279
 disk volumes
 JCL for using, 44
 SPACE DD parameter, 44
 DISKEXPT procedure, 200
 DISKGET procedure, 193, 213
 DISKMAP procedure, 192, 193, 194
 status code table, 192
 disks
 dedicated online, 193
 backing up, 198
 compressing, 193
 DISKEXPT procedure, 200
 DISKGET procedure, 213
 DISKSAVE procedure, 199
 dumpring to tape, 198
 list VTOC, 194
 listing of backups, 204
 maintenance guidelines, 194
 restore from tape, 213
 dedicated online errors, 192
 DISKMAP procedure, 194
 disks (N)
 backup, 280
 DISKSAVE procedure, 199
 DISP
 DD parameter, 37
 DISP (N)
 parameter of DD, 268, 270
 DLM (N)
 parameter of DD, 268
 DMBATCH (N), 308
 documentation. *See* publications. *See* publications
 DPRTY (N)
 parameter of EXEC, 264
 DSCOPY procedure, 107
 DSF
 not allowed, 100
 DSGET procedure, 157
 DSLIST procedure, 113
 DSN (N)
 parameter of DD, 268, 271
 DSN DD parameter, 40
 DSNAME
 DD parameter, 40
 DSNAME DD parameter
 restrictions, 40
 dsnames, 40
 duplicate, 40
 naming, 40
 DSORG DCB parameter, 34
 DSRENAME procedure, 170
 DSSAVE procedure, 156
 DSSCM procedure, 124
 DSSCMI procedure, 125
 DSSCR procedure, 171
 DSSCRV procedure, 172
 DSSUBMIT program, 226
 DSUNCM procedure, 125
 DSUNSCM procedure, 124
 DUMMY DD parameter, 42
 DUMMY DD parameter (N), 268
 duplicate names prohibited, 40

DYNAM (N)
 parameter of DD, 268
 DYNAMNBR (N)
 parameter of EXEC, 264
 EDIT format data sets
 EDSGET, 161
 EDSIN, 217
 EDSLIST, 114
 EDSOUT, 219
 EDSSORT, 120
 EDSUTIL program, 214
 read, 217
 reading and writing, 217
 utilities, 215
 write, 219
 EDIT format explained, 214
 EDSGET procedure, 161
 EDSIN procedure, 217
 EDSLIST procedure, 114
 EDSOUT procedure, 219
 EDSSAVE procedure, 159, 169
 EDSSORT procedure, 120
 EDSUTIL procedure, 135
 EDSUTIL program, 214
 efficiency of programs, 32
 encryption, 124
 utilities, 124
 encryption (N), 380
 enqueue. *See* cross section enqueue
 ENTER DISKCALC, 45
 ENTER RECOVER, 201
 ENTER RESTORE, 201
 ENTER SCRATCH, 172
 ENTER SUBMIT, 93
 erasing data (N), 324
 error messages, 241
 \$HASP185, 245
 /*MESSAGE statement in error, 253
 about JOB statement, 255
 completion codes, 15, 95, 170, 198
 contact TASC, 243
 critical, 243
 CTT108E 0545, 253
 CTTX005, 252
 disk and MSS, 247
 DISKMAP, 198
 dsnames in quotes, 250
 exceeding lines, 246
 F82, 250
 from operator, 246
 general, 244
 HASP170, 246
 HASP377, 246
 HELD FOR SYSTEM DOC, 243
 IBM messages for MSS, 249
 IBM publications, 255
 IEAP11I, 243
 IEF287I, 250
 IEF643I, 250
 IEF645I from JCLSCAN, 227
 IGD31GI, 250
 invalid data set name, 249
 multi-volume, 249
 NIH130I, 250, 252
 NIH300I, 247
 NIH801W, 247
 NIH802E, 247
 NIH813I, 247
 NIH921E, 244
 NIH984E, 245
 NIHDS06, 248
 NIHDS07, 248
 NIHDS09, 248
 NIHDS13, 249
 NIHDS27, 249
 NIHDS30, 249
 operator, 243
 OUTPUT INCOMPLETE, 245
 RJE workstation, 246
 SYSOUT DD statement, 245
 tapes, 250
 temporary data set on a public volume, 249
 UNIT=TAPE wrong, 253
 where to find, 255
 WYLBUR catastrophic, 243
 error messages (N), 329
 02I, 336
 1xx, 332
 1yy, 333
 2xx, 332
 2yy, 334

3xx, 333
 3yy, 334
 4xx, 333
 4yy, 334
 6xx, 333
 7xx, 333
 7yy, 334
 ABR, 341
 CTT143W, 252
 Dyy, 334
 Eyy, 334
 FDRW311, 341
 FDRW70, 341
 FDRW71, 341
 FDRW76, 341
 FDRW77, 341
 FDRW78, 341
 FEC, 333
 Fyy, 334
 IECTMS2, 335
 IECTMS3, 335
 IECTMS4, 336
 IECTMS6, 336
 IECTMS7, 336
 JES2 message log, 335
 PCC0001I, 337
 PCC0003I, 337
 PCC8002I, 337
 PCCX0004I, 337
 PCCX0005I, 337
 RFX204I, 338
 RFX205I, 338
 RFX206I, 338
 RFX207I, 338
 RFX208I, 338
 RFX209I, 339
 RFX210I, 339
 RFX211I, 339
 RFX299I, 339
 RFX301I, 339
 RFX302I, 340
 RFX303I, 340
 RFX308I, 340
 RFX399I, 340
 TMS, 332

estimating CPU time, 14

OS/390 Batch and Utilities (4/02)
 (N) = North System only

EXEC statement
 COND parameter, 21
 PARM parameter, 23
 PGM parameter, 20
 PROC parameter, 21
 procedure-name parameter, 21
 purpose, 18
 REGION parameter, 23
 EXEC statement (N), 263, 265
 operation field, 264
 parameters field, 264
 stepname field, 264
 execute statement (N). *See* EXEC statement (N)
 execution time, 8
 expanded JCL, 94
 Fast Dump Restore Data set Function (N).
 See FDRDSF
 FCB (N)
 parameter of DD, 272
 parameter of DD, 268
 FDRDSF (N), 298
 figures
 ADSRECOV - LIST statement
 functions, 203
 ADSRECOV - SET statement
 functions, 203
 ADSRECOV backup table capacity, 200
 ADSRECOV optional space
 parameters, 208
 data set naming standards (N), 271
 DD Parameters for Creating Data sets (N), 269
 directory of utilities, 99
 disk utilities, 150
 DISKMAP status codes, 192
 functions of PDS utilities, 172
 functions of tape utilities, 137
 general data set utilities, 105
 job turnaround phases, 8
 JOBPARM keyword parameters and values (N), 286
 maximum limits for batch jobs, 4
 other ADSRECOV parameters, 209
 phases of job turnaround, 8

print utilities, 110
 publications containing messages, 255
 resource limitations by job class (N),
 261
 SAVE and GET procedures, 153
 SYSOUT (N)
 form parameters and values, 287
 tape utilities overview, 136
 utilities (N), 293
 utilities for EDIT format data sets, 215
 file transfer
 CONNECT:Direct, 234
 host-to-host, 234, 235
 host-to-host transfer (N), 382
 to and from remotes, 126
 file transfer (N), 304
 CONNECT:Direct, 308
 from another datacenter, 307
 host-to-host, 301
 to another data center, 305
 file transfer facility (N)
 PDS, 308
 FIND DSNAMES, 130
 ADSMAP control statement, 130
 FLASH (N)
 parameter of DD, 268
 foreign tapes (N), 320
 FORMDEF
 form definition for OUTPUT statement,
 87
 FORTRAN, 225
 FREE (N)
 parameter of DD, 268, 273
 subparameters, 273
 FRGNCOPY (N), 321
 GDG. *See* generation data sets
 generation data groups, 228
 creating new, 230
 deleting, 231
 generation data sets, 228
 creating new GDG, 230
 model data set label, 229
 VDSUTIL, 231
 GROUP (N)
 parameter of JOB, 259
 HASP170, 246

header sheet, 94
 HELD messages
 for SYSTEM DOC, 243
 help, 1
 HISTORY DD statement
 use with DISKMAP, 197
 HOLD
 output HOLD queue, 82
 HOLD (N)
 parameter of DD, 273
 parameter of DD, 268
 host-to-host file transfer, 234
 host-to-host file transfer (N), 301, 382
 IBM standard dsnames, 63
 IBM utilities
 DFSORT, 117
 IEBCOMPR, 138
 IEBGENER, 105, 138, 141
 IEBTPCH, 138, 141
 IEBUPDATE, 138
 IEHPROGM, 230
 IBM utilities (N), 293
 ICAPRTBL
 not allowed, 100
 IDCAMS (N), 294
 IEBCOMPR, 101, 138
 IEBCOPY, 101
 IEBCOPY (N), 293
 IEBDG, 101
 IEBDG (N), 293
 IEBEDIT, 101
 IEBGENER, 101, 105, 138, 141
 IEBGENER (N), 294
 IEBIMAGE
 not allowed, 100
 IEBISAM, 101
 IEBTPCH, 101, 138, 141
 IEBTPCH (N), 294
 IEBUPDTE, 101, 138
 IEBUPDTE (N), 294
 IEHATLAS
 not allowed, 100
 IEHINITT
 not allowed, 100
 IEHLIST
 restriction, 101

IEHMOVE
 restriction, 101
 IEHPROGM, 230
 restriction, 101
 IEHPROGM (N), 294
 IFHSTATR
 not allowed, 100
 INCLUDE groups, 71
 input stream data set parameter (N), 269
 instream procedures, 70
 integrity of data sets, 72
 interactive job submissions (N), 292
 interactive processing (N)
 canceling jobs, 292
Interface, 1
 introduction, 1
 ISAM data sets
 not supported, 35
 ISPF (N)
 ABR, 282
 JCL
 comment statement, 91
 DD statement, 24
 do not copy, 66
 EXEC statement, 18
 expanded, 94
 explained, 4
 functions, 2, 3
 JCLLIB, 66, 71
 JCLSCAN procedure, 227
 job dependency control statements, 72
 job execution control statements, 77
 JOB statement, 11
 output control statements, 81
 OUTPUT statement, 86
 printing control statements, 90
 resource control statements, 75
 restricted parameters, 6
 scanning for errors, 227
 syntax error checking, 4
 temporary data sets, 61
 Titan, 1, 3, 257
 JCL (N)
 DD statement, 265
 EXEC statement, 263
 JCLLIB statement, 263
 OUTPUT statement, 277
 overview, 257
 JCL libraries, 66
 JCL statements. *See also* DD statement
 control access to master file, 73
 hold output, 81
 INCLUDE groups, 71
 mount a preassigned tape, 75
 prevent jobs running simultaneously, 73
 print held output, 88
 reduce costs, 78
 route output to EPS printers, 81
 route output to central facility, 81
 route output to RJE workstations, 81
 sent to other computer nodes, 81
 title on output, 90
 JCLLIB, 66
 JCLLIB statement, 71
 JCLLIB statement (N), 263
 JCLSCAN procedure, 227
 JES2 (N), 284
 JES2 control statements
 /*ACCESS, 77
 /*AFTER, 73
 /*ASSIGN, 77
 /*CNTL, 73
 /*DDOUT, 84
 /*DISCOUNT, 78
 /*JOBOUT, 82
 /*KEYWORD, 78
 /*MESSAGE, 75
 /*NETACCT, 81
 /*NOPURGE, 88
 /*NOTIFY, 78
 /*PASSWORD, 80
 /*PRIORITY, 79
 /*QUICK, 79
 /*RERUN, 80
 /*ROUTE, 81
 /*ROUTE XEQ, 76
 /*ROUTE XEQ LATE, 81
 /*TITLE, 90
 /*XMIT, 80
 JES2 control statements (N), 284
 /*AFTER, 284
 /*BEFORE, 284

- /*BOX, 284
- /*JOBPARM, 285
- /*NETACCT, 286, 302
- /*OUTPUT, 286
- /*PASS, 288
- /*PRINT, 288
- /*PRIVATE, 289
- /*ROUTE, 289, 303
- /*XMIT, 291, 302
- JES2 job log, 94
- JES2 message log (N), 335
- JES2 RJE control statements (N)
 - /*SIGNOFF, 291
 - /*SIGNON, 291
- job accounting block, 95
- job cancellation (N), 292
- job class order, 74
- job class standards, 9
 - Class A, 9
 - Class B, 10
 - Class C, 10
 - Class E, 9
 - class F, 11
 - class H, 11
- job class standards (N), 261
- job classes (N), 358
- job control language. *See* JCL
- Job Entry Subsystem. *See* JES2
- job output. *See* output
- job standards, 4
- JOB statement, 11, 13
 - accounting information, 14
 - complete form, 13
 - COND parameter, 15
 - error messages listed, 255
 - MSGLEVEL parameter, 16
 - REGION parameter, 16
 - RESTART parameter, 17
 - simple explanation, 12
 - USER parameter, 18
- JOB statement (N), 257
- JOB statement parameters (N)
 - accounting information, 258
 - operation field, 258
 - optional parameters, 259
 - user identification field, 258
- job submission (N), 292
 - interactive, 292
 - RJE, 292
 - Web, 292
- job turnaround
 - Class A, 9
 - Class B, 10
 - Class C, 10
 - Class E, 9
 - class F, 11
 - class H, 11
 - delays, 91
 - delays Awaiting Execution, 91
 - delays due to data set contention, 92
 - delays due to tape handling, 91
 - delays long Elapsed Time, 92
 - delays when /*AFTER used, 91
 - delays when /*BEFORE used, 91
 - delays when /*CONTROL used, 91
 - delays when /*DISCOUNT used, 91
 - delays when /*ROUTE XEQ LATE used, 91
 - delays when BDAM used, 92
 - delays when RECOVER is used, 91
 - delivery times, 8
 - execution turnaround, 8
 - explained, 8
 - increased by /*AFTER, /*BEFORE, /*CNTL, 72
 - output turnaround, 8
 - phases, 8
- JOBPARM (N)
 - keyword parameters, 286
- jobs
 - execution order, 73
 - output explained, 94
- keyword parameters (N), 286
- keywords
 - protection, 78
- LABEL, 43
- LABEL (N)
 - parameter of DD, 268, 273
 - subparameters, 273
- LABEL DD parameter, 43
- LABEL subparameters, 43
- LATE resource, 76, 81

level of support, 1
LIKE DD parameter, 53
limits. *See* restrictions, job standards
lines per job
 includes copies, 14
 maximum, 4
 specifying, 14
LISTPDS program, 186
Loader
 functions described, 225
lost disk data sets
 ADSRECOV
 private volumes, 201
LRECL, 31
magnetic tapes. *See* tapes
management class
 how to change, 103
manuals. *See* publications
MARKCON (N), 326
MARKDUMP (N), 326
MARKIV (N), 326
MARKIVD (N), 326
MARKREST (N), 326
MARKUTIL (N), 326
MAX (N), 294
MAX BATCH (N), 295
MAX DATA/UTIL (N), 294
MAX/PDF (N), 295
MEMADD procedure, 176
MEMALTER procedure, 178
MEMRENAM procedure, 177
MERGE procedure, 122
merging data sets
 utilities for, 117, 122
messages, 241. *See also* error messages
 where to find, 255
messages (N), 329
 ABR, 341
 common ABENDs, 329
 error messages
 RACF, 336
 JES2 message log, 335
 RACF, 336
 TMS codes and messages, 332
microfiche
 RESPOUT procedure, 143

utility to obtain more output, 143
Model 204 (N), 376
MODIFY (N)
 parameter of DD, 268
modifying EXEC parameters, 68
money. *See* charging
MSGCLASS (N)
 parameter of JOB, 259
MSGLEVEL (N)
 parameter of JOB, 259
MSGLEVEL parameter
 JOB statement, 16
MSS
 cataloging data sets, 101
 DISP parameter, 37
 scratching data sets, 171
naming conventions
 cataloged data sets, 102
 data sets, 40
 IBM standard, 40
 VSAM data sets, 40
NDM (N). *See* CONNECT:Direct
Network Data Mover (N). *See*
 CONNECT:Direct
Network Job Entry. *See* NJE
Network Job Entry (N). *See* NJE (N)
NIH Computer Center User's Guide, 1
NIH Data Warehouse, 228
NIH error messages, 244
NJE, 80, 234
NJE (N), 301, 382
node names (N)
 host-to-host file transfer, 301
North System. *See* Titan
NOTIFY (N)
 parameter of JOB, 259
NOTIFY attribute, 4
nullifying symbolic parameters, 68
OLS (N). *See* Online SyncSort
Online SyncSort (N), 298
operating system, 2, 3
 functions, 3
 information, 94
 overview, 3
operation field (N), 264, 267
OPTCD DCB parameter, 34

order
 job execution, 74
order of execution, 73
OS/390 South System. *See* South System
Output Stream Data set (N), 275
OUTLIM (N)
 parameter of DD, 268
output
 changing lines per job, 14
 turnaround time, 8
 understanding, 94
output control statements
 direct output, 81
output directed to another computer node,
 81
OUTPUT HOLD queue
 /*NOPURGE, 88
 disposition of jobs in, 82
OUTPUT JCL statement (N), 278
OUTPUT statement
 examples, 87
 for use with cut-sheet printer, 86
 required parameters, 87
OUTPUT statement (N), 277
 parameters field, 278
 stepname field, 278
overview, 3
page size, 83
paging
 JCL to control, 83, 85
PALTape utility, 141
parameters field (N), 264, 267
Parklawn site (N), 323
PARM (N)
 parameter of EXEC, 265
PARM parameter, 23
partitioned data set. *See* PDS
PASBILL procedure, 228
passing data sets, 65. *See* DD statements
PASSWORD (N)
 parameter of JOB, 259
PCCDCODS (N), 315
PCCNCODS (N), 315
PCCVTOC (N), 301
PDS
 add members, 176
 allocate space, 174
 cannot be scrambled, 124
 condense, 181
 copy entire disk data sets, 182
 copy members, 184
 delete members, 175
 list contents, 180
 list member names, 180
 list members, 186
 modify existing load module, 178
 procedures to handle, 172
 rename a member, 177
 unload entire, 163
 utilities to manage, 172
PDS (N)
 file transfer, 308
 rules, 263
PDSALLOC procedure, 174
PDSCOPY, 182
PDSDEL procedure, 175
PDSGET procedure, 164
PDSLST procedure, 186
PDSMAP procedure, 180
PDSPACK procedure, 181
PDSSAVE procedure, 163
 hints, 170
PERFORM (N)
 parameter of EXEC, 265
 parameter of JOB, 259
performance
 how to improve, 32
PGM (N)
 parameter of EXEC, 264
PL/I, 225
PostScript printing, 116
POSTSCRIP procedure, 116
PRINT procedure, 110
printing
 character sets, 85
 combining output control statements, 88
 costs, 96
 data sets, 109
 disk data sets, 186
 ejecting, 83, 85
 JCL to control, 82, 90
 OUTPUT statement, 86

output directed to another computer
 node, 81
 page size
 /*DDOUT to specify, 84
 PostScript, 116
 procedures, 109, 186
 utilities for, 109
 printing changes (N), 373
 priority of jobs, 79
 listed for job classes, 79
 privacy. *See* security
 private disks. *See* disks, dedicated online
 private procedure library (N), 263
 PROC
 parameter of EXEC, 21
 PROC (N)
 parameter of EXEC, 264
 procedure libraries, 66
 procedure-name parameter, 21
 procedures. *See* cataloged procedures
 do not copy, 66
 utilities, 99
 Program Manager Binder, 222
 program output, 96
 PROTECT (N)
 parameter of DD, 268
 PRTY (N)
 parameter of JOB, 259
 public disk storage (N)
 ABR, 281
 public volumes
 data sets, 40
 PUBLIC.BACKUPS. *See* &PUBLIC
 BACKUPS
 publications, 1, 100
 ordering, 2
 ordering service, 2
 purging jobs, 82
 QNAME (N)
 parameter of DD, 269
 RACF
 changes(N), 368
 SHOW DSNAMES option of
 ADSMAP messages, 132
 user parameter, 18
 RACF (N)
 messages, 336
 RACF changes(N), 369
 RACFREG (N), 379
 rates. *See* charging
 RCVFILE, 234, 235
 RCVFILE (N), 304, 382
 RD (N)
 parameter of EXEC, 265
 parameter of JOB, 259
 recalling data sets via batch, 170
 RECFM, 31
 recovering lost data sets
 for private volumes, 201
 reel drives (N)
 foreign tape processing, 323
 REGION (N)
 parameter of EXEC, 265
 parameter of JOB, 259
 REGION parameter, 13
 EXEC statement, 23
 JOB statement, 16
 region size
 estimating, 223
 remote job entry. *See* RJE workstations
 renaming disk data sets, 170
 renaming multiple data sets, 103
 resource control statements
 tape and disk, 75
 resource limitations by job class (N)
 table, 261
 RESPOUT procedure, 143
 RESTART
 JOB statement, 17
 RESTART (N)
 parameter of JOB, 260
 restart points
 required, 10
 restore from tape, 193
 restoring dedicated disks, 213
 restrictions
 DCB subparameter, 34
 DCB subparameters, 30
 DSORG subparameter, 34
 JCL parameters, 6
 LABEL parameter, 43
 no unmoveable data sets, 30, 34

on utilities, 100
tapes, 5
return codes, 15, 95, 169
RJE control statements (N), 291
RJE workstations
transmitting data to/from tapes, 126
RJE workstations (N)
canceling jobs, 292
submitting jobs, 292
RMTDSIN procedure, 126
RMTDSOUT procedure, 126, 128
ROUTE XEQ statement (N), 289
SANITIZE (N), 324
scrambling data sets, 124
scratch packs (N), 279
scratch space limit, 4
scratching
disk data sets, 171
multiple disk data sets, 103, 172
single disk data sets, 171
Secure Internet LinKed technologies. *See* SILK
security
/*KEYWORD, 78
disk data set backups, 201
encryption, 124
scrambling data sets, 124
security (N)
tapes, 312
SENDFILE, 234, 235
SENDFILE (N), 304, 382
Service Request Ticket. *See* SRT
SHOW DSNAMES, 132
SILK
batch job submission, 92
SILK (N)
batch job submission, 292
tape inventory management, 317
software
improving efficiency, 32
restricted IBM utilities, 100
software dropped (N), 359
software facilities. *See* utilities
SORT procedure, 119
SORT/MERGE/COPY (N), 296
sorting
396

utilities, 117
SPACE (N)
parameter of DD, 269, 275
SPACE DD parameter, 44
space management (R), 280
special tapes, 149
SRT, 2
standards
job classes, 9
restricted utilities, 100
standards (N)
tapes, 316
standards for batch jobs. *See* job standards
STEPLIB DD statement, 66
stepname field (N), 264
storage (N)
disk, 279
public storage user packs (N), 280
scratch packs, 279
SUBSYS (N)
parameter of DD, 269
SUPERZAP
use MEMALTER, 178
SUPRVTOC procedure
replaced by DISKMAP, 194
symbolic parameters
assigning values, 68
nullifying, 68
SyncSort - online (N), 298
SYNCSORT (N), 296, 297
SYSOUT (N)
form parameters and values, 287
parameter of DD, 269, 275
subparameters, 276
SYSOUT DD
described, 28
SYSOUT DD statement
error message, 245
system catalog
described, 101
how to use, 101
naming restrictions, 102
SYSTEM DOC
hold messages, 243
tape changes (N), 364
tape data sets, 43

cataloging, 101
creating and using, 61
creating permanent, 63
creating temporary, 63
JCL, 43
JCL for using, 40
JCL statements for using, 64
limit, 64, 169
multi-volume, 51
naming conventions, 40
uncataloging, 101
utilities to handle, 105, 135
tape data sets (N)
 recreating, 317
tape inventory management (N)
 via Web, 317
tape library, 319
 storage (N), 310, 319
Tape Management System (N). *See* TMS
Tape Security System (N). *See* TSS
TAPECOPY program, 142
TAPEMAP procedure, 139
tapes. *See also* DD statements
 /* MESSAGE statement to mount
 preassigned tapes, 75
 /* ACCESS, 77
 /* ASSIGN, 77
 /* ROUTE statements, 76
 9-track 6250 BPI, 198
 ASCII data on, 137, 142
 DCB, 36
 density, 36
 device names generic, 102
 generic device, 102
 mount message, 75
 multi-data set, 169
 not in tape library, 75
 PALTAPE utility, 141
 releasing, 147
 RESPOUT procedure, 143
 restrictions, 5
 sharing tapes, 77
 special tapes information, 149
 TAPECOPY, 142
 TAPEMAP utility, 139
 TAPESCAN procedures, 149
 transmitting data via RJE workstations,
 126
 uncataloged when released, 147
 UNIT DD parameter, 46
 utilities, 135
 functions, 136
 utilities summary, 136
tapes (N), 310
 access list, 313, 314
 data sets, 310
 expiration dates, 310
 foreign, 320
 foreign reel tapes, 323
 inventory reminders, 318
 purchase, 319
 reel drives, 323
 removal, 319, 320
 standards, 316
 TMS error conditions, 312
 TMS/CA1, 310
TAPESCAN procedure, 149
TASC, 1, 2
Technical Assistance and Support Section.
 See TASC
temporary data sets, 61
TERM (N)
 parameter of DD, 269
ThruPut Manager (N), 358
TIME (N)
 parameter of EXEC, 265
 parameter of JOB, 260
Titan, 1
 introduction, 1
 JCL, 1, 3, 257
 transition, 99
 transition (N), 293
 WYLBUR transition, 3, 231
TMS (N), 310
 completion codes and messages, 332
 JES2 message log, 335
 messages, 332
trailer sheet, 96
transition Web pages, 344
TSS (N), 312
turnaround. *See* job turnaround
TYPRUN (N)

parameter of JOB, 260
 UCS (N)
 parameter of DD, 269
 uncataloging data sets on released tapes,
 147
 UNCATDS
 procedure, 103
 UNIT (N)
 parameter of DD, 269, 276
 UNIT DD parameter, 46
 unscrambling data sets, 124
 USER
 JOB statement parameter, 18
 USER (N)
 parameter of JOB, 260
 user identification field (N), 258
 user packs (N), 280
 USRPDSC procedure, 184
 USRPDSL procedure, 180
 utilities
 ADSERASE, 133
 ADSMAP, 130
 ADSRECOV, 200, 201
 catalog a data set, 102
 CATDS, 102
 COMPARE, 123
 compare data sets, 123
 CONNECT:Direct, 238
 COPY, 105
 copy multi-data set tape, 142
 DDSCOPY, 190
 DDSGET, 167
 DDSLIST, 189
 DDSSAVE, 166
 disk data sets, 105, 149
 disk to tape, 152
 DISKEXPT, 200
 DISKGET, 193, 213
 DISKMAP, 192, 193, 194
 DISKSAVE, 199
 DSCOPY, 107
 DSGET, 157
 DSLIS, 113
 DSRENAME, 170
 DSSAVE, 156
 DSSCM, 124

DSSCMI, 125
 DSSCR, 171
 DSSCRV, 172
 DSSUBMIT, 226
 DSUNSCM, 124
 DSUNSCMI, 125
 dump tape labels, 139
 EDIT format data sets, 214
 EDSGET, 161
 EDSIN, 217
 EDSLIS, 114
 EDSOUT, 219
 EDSSAVE, 159
 EDSSORT, 120
 EDSUTIL, 135, 214
 functions, 150
 generation data groups, 228
 handling PDSs, 172
 introduction, 1
 JCLSCAN, 227
 list of, 99
 listing a bill, 228
 MEMADD, 176
 MEMALTER, 178
 MEMRENAM, 177
 MERGE, 122
 merging data sets, 117, 122
 modify existing load module, 178
 not permitted, 100
 PALTAPE, 141
 PASBILL, 228
 PDSALLOC, 174
 PDSCOPY, 182
 PDSDEL, 175
 PDSGET, 164
 PDSLIS, 186
 PDSMAP, 180
 PDSPACK, 181
 PDSSAVE, 163
 POSTSCR, 116
 PRINT, 110
 printing a data set, 110
 printing disk data sets, 186
 printing hexadecimal, 112
 RCVFILE, 234, 235
 recover private disk backups, 201

- renaming data sets, 170
- RESPOUT, 143
- restrictions, 100
- RMTDSIN, 126
- RMTDSOUT, 126, 128
- scan JCL, 227
- scramble data sets, 124
- scratching disk data sets, 171
- scratching multiple data sets, 172
- SENDFILE, 234, 235
- SORT, 119
- sorting data sets, 117
- submit batch jobs, 226
- summary for EDIT format, 214
- summary for handling data sets, 105
- summary for printing, 110
- tape data sets, 105, 135
- TAPECOPY, 142
- TAPEMAP, 139
- tapes, 135
- tapes with problems, 141
- TAPESCAN, 149
- UNCATDS, 103
- unscramble data sets, 124
- USRPDSC, 184
- USRPDSL, 180
- VDSUTIL, 103, 230, 231
- VOLSTAT, 144
- WYLBUR, 232
- WYLBURL, 232
- utilities (N), 293
 - COPY. *See* SYNCSORT
 - DMBATCH, 308
 - FDRDSF, 298
 - FRGNCOPY, 321
 - IBM, 293
 - IEBCOPY, 293
 - IEBDG, 293
 - IEBGENER, 294
 - IEBTPPCH, 294
 - IEBUPDTE, 294
 - IEHPROGM, 294
 - MARKCON, 326
 - MARKDUMP, 326
 - MARKIV, 326
 - MARKIVD, 326
 - MARKREST, 326
 - MARKUTIL, 326
 - MAX, 294
 - MAX BATCH, 295
 - MAX DATA/UTIL, 294
 - MAX/PDF, 295
 - MERGE. *See* SYNCSORT
 - Online SyncSort, 298
 - PCCVTOC, 301
 - RCVFILE, 304
 - SANITIZE, 324
 - SENDFILE, 304
 - SORT. *See* SYNCSORT
 - SORT/MERGE/COPY, 296
 - SYNCSORT, 296
 - SYNCSORT, 297
- utilities list, 99
- utilities summary
 - disk data sets, 150
- utility programs. *See* utilities
- VDSUTIL procedure, 103, 230, 231
- VOL (N)
 - parameter of DD, 269
- VOLSTAT procedure, 144
- VOLUME DD parameter, 49
- volume table of contents. *See* VTOC
- VSAM, 35
 - data set information, 130
- VSAM data sets
 - DSRENAME restriction, 170
 - naming conventions, 40
- VSAM files (N), 294
- VTOC, 192, 200
- Web Sponsor (N), 379
- World Wide Web
 - batch job submission, 92
 - publications, 2
- World Wide Web (N)
 - canceling jobs, 292
 - job submission, 292
- WYLBUR. *See also* Edit format data sets
 - /*DATA, 90
 - /*IBM, 90
 - /*INSERT, 90
 - /*MERGE, 90
 - /*NUMBERED, 90

`/*OVERLAY`, 90
`/*TSO`, 90
`/*UNNUMBERED`, 90
commands executed in batch, 232
`ENTER PDSLIST`, 186
`ENTER RECOVER`, 201

`ENTER RESTORE`, 201
`ENTER SCRATCH`, 172
`ENTER SUBMIT`, 93
transition, 3, 231
`WYLBUR` procedure, 232
`WYLBURL` procedure, 232

OS/390 Batch and Utilities

Document Evaluation

Is the Manual:

	YES	NO
Clear?	<input type="checkbox"/>	<input type="checkbox"/>
Well organized?	<input type="checkbox"/>	<input type="checkbox"/>
Complete?	<input type="checkbox"/>	<input type="checkbox"/>
Accurate?	<input type="checkbox"/>	<input type="checkbox"/>
Suitable for the beginner?	<input type="checkbox"/>	<input type="checkbox"/>
Suitable for the advanced user?	<input type="checkbox"/>	<input type="checkbox"/>

Comments:

Please give page references where appropriate. If you wish a reply, include your name and mailing address.

Send to: National Institutes of Health
 Center for Information Technology
 NIH Computer Center
 Building 12A, Room 4011
 12 South Drive MSC 5607
 Bethesda, MD 20892-5607

FAX to: (301) 496-6905

IC or Agency:
Date Submitted:
Name (Optional):
E-mail Address:



DEPARTMENT OF
HEALTH AND HUMAN SERVICES
PUBLIC HEALTH SERVICE

National Institutes of Health
Building 12A, Room 1011
12 South Drive MSC 5605
Bethesda, MD 20892-5605

