COS™ Operational Procedures Reference Manual

SM–0043 G

Cray Research, Inc.

The UNICOS operating system is derived from the AT&T UNIX System V operating system. UNICOS is also based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California.

Requests for copies of Cray Research, Inc. publications should be sent to the following address:

Cray Research, Inc.
Distribution Center
2360 Pilot Knob Road
Mendota Heights, MN 55120

Order desk (612) 681–5907
Fax number (612) 681–5920

NEW FEATURES

This reprint with revision of the COS Operational Procedures Reference Manual supports COS release 1.17.1. This release includes the automated tape loader and new disk support.

The automated tape loader, with support of other front-end software, can identify, mount, and unload 3480-style tapes without operator intervention. Initially, two loaders are supported: Storage Technology 4400 Nearline and IBM Scratch Loader. This feature requires supporting changes to the VM 6.0 and MVS 3.01 stations. See section 2 for more information.

RECORD OF REVISION

CRAY RESEARCH, INC.

PUBLICATION NUMBER SG-0043

Each time this manual is revised and reprinted, all changes issued against the previous version are incorporated into the new version and the new version is assigned an alphabetic level.

Every page changed by a reprint with revision has the revision level in the lower righthand corner. Changes to part of a page are noted by a change bar in the margin directly opposite the change. A change bar in the margin opposite the page number indicates that the entire page is new. If the manual is rewritten, the revision level changes but the manual does not contain change bars.

Requests for copies of Cray Research, Inc. publications should be directed to the Distribution Center and comments about these publications should be directed to:

CRAY RESEARCH, INC.
1345 Northland Drive
Mendota Heights, Minnesota 55120

Revision	Description
	October 1980 - Original printing; this publication supports COS Version 1.09. This obsoletes (in general) Parts 2 and 3 of the CRAY-OS System Programmer's Manual, publication 2240012. This publication represents a complete rewrite, and changes are not noted by change bars.
A	July 1981 - Reprint with revision. This reprint includes descriptions of the following: changes in the generation of the system; the generation of new programs, MODSEQ, FLODUMP, and SYSREF; the procedures for establishing a system bulletin; new startup parameter file directives that delete previously specified flaws, suppress the recovery of the $SYSTEMLOG buffer, and force a new edition of $SYSTEMLOG; and other miscellaneous technical changes to bring this publication into agreement with the 1.10 version of COS software. All previous printings are obsolete.
A-01	June 1982 - Change packet. This printing includes the following additions: examples of COS software generation using the procedural library, ITEMIZE, TEDI, and SID program generation, new COS software procedure parameters, generation steps for Data General station software, revised configuration examples, on-line magnetic tape configuration, OPTION procedures, tape device control, *SKIPEFT directive, new sample jobs in Appendix A, new installation parameters, and the addition of Appendixes D, E and F. Miscellaneous technical and editorial changes are also included to bring the manual into agreement with the 1.11 version of COS software.
B	August 1982 - Reprint with revision. This printing incorporates change packet A-01. No other changes have been made.
C	July 1983 - Rewrite. Major new features documented include a new job scheduler, new program library organization, startup reconfiguration and error recovery, permanent dataset privacy and security, and the Fast Secondary Storage Subsystem.

Miscellaneous technical and editorial changes are also included to bring the manual into agreement with the 1.12 version of COS software.

A section has been added on dumping the Cray computer system, and a section on adding a task has been moved to the COS EXEC/STP/CSP Internal Reference Manual, CRI publication SM-0040. Specific, detailed system generation procedures and sample COSgen jobs formerly in this manual are now on the COS 1.12 release tape. Detailed installation information for COS 1.12 is now in the System Installation Bulletin.

C-01 October 1983 - Change packet. This printing includes procedures which were previously identified as being deferred and are now implemented (generation procedures for Pascal ($PSCLIB and PASCAL) and SEGLDR (SEGLDR and SEGRLS) and the restore process for fast secondary storage, including *RESTORE and dataset restoration error messages), a description of the datasets generated by GCOS and GIOS, and the parameter lists for the miscellaneous generation procedures (GENTAPE, GETBAS, GETPLS, OLDSEQ, NEWSEQ and PURGE). Changes have been made to the privacy system, the Link Configuration Table (LCT), and to the *DXTERR directive. Lists of the UPDATE decks used for CSIM and IOPPL are now included in Appendix C.

C-02 February 1984 - Change packet. This printing includes the following features: movement of several utilities from COSPL to UTILPL, Data General Station generation procedure changes, multitasking, generic and controlled devices, COS now released configured for a CRAY-1 M Series computer system, striped disk groups, a new macro (LDEV), and numerous installation parameter changes.

Miscellaneous technical and editorial changes are also included to bring the manual into agreement with the 1.13 version of COS software.

C-03 December 1984 - Change packet. This change packet reflects feature changes made to COS for the 1.14 release, including Integrated Support Processor (ISP) and X-MP/4 support. Note that the completed ISP code is not available; you will be notified of its availability by letter. Section 2 has been rewritten to include specific examples and procedures for generating site-specific COS software. Startup permanent dataset recovery and hold directives have been removed and messages updated. Rolled job recovery and system directory processing have been added. Installation parameters in Appendix B have been organized into functional groups, with the Solid-state Storage Device (SSD) parameters added.

D January 1986 - This reprint with revision brings the manual into agreement with version 1.15 of COS software. The following features were added: permanent dataset archiving, bypass label processing, CONFIG enhancements, DD-39 disk drive support, Disk-queued I/O, DSC error recovery, large memory scheduling, new expander chassis support, moving NOTE to TOOLPL, 3480 tape support, automatic volume recognition, and on-line diagnostics. The Guest Operating System (GOS) directive has been added, running the Cray operating system UNICOS. The Integrated Support Processor (ISP), formerly deferred, has been implemented. Appendix A on Maximum Memory Algorithm has been removed. This rewrite includes miscellaneous technical and editorial changes. This manual obsoletes all previous printings.

E April 1987 - This rewrite brings the manual into agreement with version 1.16 of COS software. The following features were added: SUPERLINK, new SYSDFPL, memory error correction, FSS enhancements, device partitioning, and disk flaw directive enhancements. This rewrite includes a reorganization of sections 1, 2, and appendix A, along with miscellaneous editorial and technical changes throughout the rest of the manual.

E-01 September 1987 - This change packet supports the CRAY X-MP/14se computer system and also includes miscellaneous editorial changes.

F October 1988 - This reprint with revision supports the COS 1.17 release. Major features include support of the CRAY Y-MP computer system, Resource Dataset Management (RDM), demand processing, and user exits.

G November 1989 - This reprint with revision supports the COS 1.17.1 release.

PREFACE

This manual describes the procedures for generating, installing, and debugging Cray operating system COS software for use on the CRAY Y-MP, CRAY X-MP, CRAY X-MP EA, and CRAY-1 computer systems. The manual deals with the following aspects of COS:

- Software generation. Section 1 describes the programs and procedures for generation of COS, I/O Subsystem (IOS), and Data General Station (DGS) software. Section 2 discusses configuration of COS software to your specific site. Appendix B lists all UPDATE program libraries.

- COS installation. Section 3 describes the software release process and the installation and implementation of COS (including security, dataset privacy, and archiving). Section 4 covers the Job Scheduler; section 5 discusses startup and shutdown operations. Appendix A lists installation parameters and defaults.

- COS debugging and dumping. Sections 6 and 7 describe the debugging and dumping of the Cray computer systems.

Operator commands and maintenance control utilities are not described in this manual; see the Data General Station (DGS) Operator's Guide, Cray Research, Inc. (CRI) publication SG-0006, or the I/O Subsystem (IOS) Operator's Guide for COS, CRI publication SG-0051, for these details. For startup procedures for a Cray computer system with an IOS, see the I/O Subsystem (IOS) Operator's Guide for COS. For additional IOS information, see the I/O Subsystem (IOS) Administrator's Guide, CRI publication SG-0307.

This manual is part of a set of manuals that describes the internal design of COS and its product set. Other manuals in the set are as follows:

SM-0007	IOS Table Descriptions Internal Reference Manual
SR-0012	Macros and Opdefs Reference Manual
SR-0013	UPDATE Reference Manual
SM-0017	Fortran (CFT) Internal Reference Manual
SQ-0023	COS Ready Reference Manual
SM-0042	Front-end Protocol Internal Reference Manual
SM-0045†	COS Table Descriptions Internal Reference Manual

† This manual is distributed on magnetic tape and can be obtained through your CRI representative.

SM-0046	IOS Software Internal Reference Manual
SM-0049	Data General Station (DGS) Internal Reference Manual
SM-0072	Cray Simulator (CSIM) Internal Reference Manual
SM-0075	COS Table Diagram Generator (TDG) Reference Manual
SD-0127	COS Accounting Aids Internal Reference Manual
SM-0140	COS Internal Reference Manual Volume I: EXEC
SM-0141	COS Internal Reference Manual, Volume II: STP
SI-0180	SUPERLINK MVS and COS Installation, Tuning, and Customization Guide
SN-3033	COS Autotasking User's Guide

Manuals defining procedures and external features of tools needed for installing and maintaining CRI software are as follows:

SM-0044	Operational Aids Reference Manual
SR-0073	Cray Simulator (CSIM) Reference Manual

Readers of this manual should be familiar with the contents of the COS Reference Manual, CRI publication SR-0011, and experienced in coding the Cray Assembly Language (CAL) as described in the CAL Assembler Version 1 Reference Manual, CRI publication SR-0000.

Other related CRI manuals that may help users are as follows:

HR-0029	CRAY-1 S Series Mainframe Reference Manual
HR-0030	I/O Subsystem Model B Hardware Reference Manual
SR-0039	COS Message Manual
HR-3005	CRAY X-MP Computer Systems Functional Description Manual

All manuals referenced throughout this publication are CRI manuals unless specifically noted otherwise.

CONVENTIONS

The following conventions are used throughout this manual unless indicated otherwise:

Convention	Description
Italics	Define generic terms representing the words or symbols to be supplied by the user
[ ] Brackets	Enclose optional portions of a command format
Choice 1 Choice 2	Indicates two or more literal parameters when only one choice can be used
<u>UN</u>DERLINE	Indicates the abbreviated form of a command

Many of the macro definitions used throughout this manual extend beyond a single CAL line. These multiline macros should be properly formatted as CAL continuation lines. See the CAL Assembler Version 1 Reference Manual for more information on CAL continuation lines.

READER COMMENTS

If you have any comments about the technical accuracy, content, or organization of this manual, please tell us. You can contact us in any of the following ways:

- Call our Technical Publications department at (612) 681-5729.
- Send us electronic mail from a UNICOS or UNIX system, using one of the following electronic mail addresses:

  uunet!cray!publications

  sun!tundra!hall!publications

- Send us electronic mail from a UNICOS or UNIX system, using the following ARPAnet address:

  publications@cray.com

- Send a facsimile of your comments to the attention of "Publications" at fax number: (612) 681-5602.
- Use the postage-paid Reader Comment form at the back of this manual.
- Write to us at the following address:

  Cray Research, Inc.
  Technical Publications Department
  1345 Northland Drive
  Mendota Heights, Minnesota 55120

We value your comments and will respond to them promptly.

2. GENERATING SITE-SPECIFIC COS SOFTWARE (continued)

APPENDIX SECTION

B. UPDATE PROGRAM LIBRARIES (continued)

FIGURES

TABLES

1. COS AND DATA GENERAL STATION (DGS) SOFTWARE GENERATION

This section introduces you to the COS software generation procedures and the DGS software.

This section describes the COS software generation process in general terms. The Release Notice, which is included with every software release, contains the specific procedures you need to generate and install that specific release. You should use this section along with the Release Notice when you perform the COS software generation and installation. The UPDATE Reference Manual, publication SR-0013, which shows you how to modify the source code, will also be helpful when you actually generate the COS software.

This section is divided into the following subsections:

- Introduction to COS software generation
- GENPL
- Generation Jobs
- GENPL Generation Procedures
- Data General Station Software Generation

The first three subsections explain the generation of COS software. The last subsection covers the generation of the Data General Station (DGS) software.

1.1 INTRODUCTION TO COS SOFTWARE GENERATION

COS software generation consists of converting and modifying the source code contained in various program libraries (PLs) into executable binaries. See the Release Notice for specific procedures for generating the COS software. During the generation process, you may also generate listings (e.g., compiler listings and load maps) that will help you with further generations and modifications. Figure 1-1 shows an overview of the generation process.

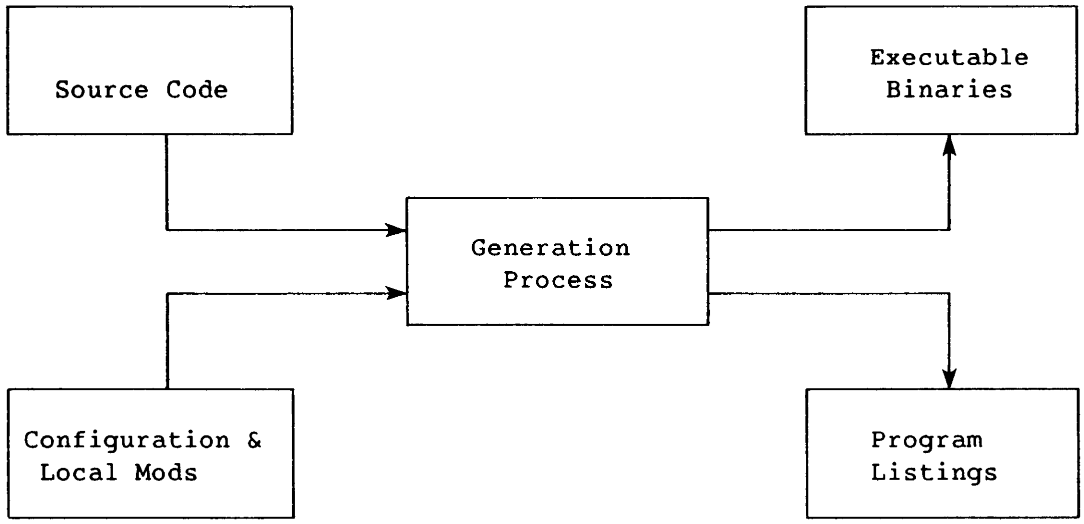

Figure 1-1. COS Software Generation Process

One of the PLs released with COS is GENPL. GENPL contains the source code for the generation process. It consists of jobs and JCL procedure definitions. These jobs and procedures act as a higher-level interface to the generation of COS software. In this way, the system analyst is isolated from the many lower-level details.

Software dependencies and the propagation of configuration parameters require a particular build order (e.g., texts before libraries, libraries before compilers, etc.). The generation jobs and procedures ensure that the software is built in the correct order.

1.1.1 ORGANIZATION OF GENPL

GENPL contains a series of decks that comprise all the procedures and jobs that generate the COS products. The first deck in GENPL is the common deck GENPLDOC. This common deck contains a summary of the information in this chapter. Following GENPLDOC are the product sections. Each product section contains the procedures and jobs associated with the generation of a particular product. The following is a list of the product sections in the order that they are found in GENPL, along with their GENPL name:

GENPL Name	Product
AUT†	Autotasking preprocessor and midprocessor
C†	C compiler, preprocessor and library.
CFT†	CFT compiler.
C77†	CFT77 compiler.
COS	COS, COS libraries and COS utilities.
CSM	CSIM utility.
DIA	On-line diagnostics.
IOS†	IOS software.
LIB	Other libraries.
MIG†	COS migration tools
MSC	Miscellaneous GENPL jobs and procedures.
PSC†	PASCAL compiler and library.
PRD	Other products.
SL	SUPERLINK software.

Each product section within GENPL is delineated by dummy common decks named BGN-*name* and END-*name*, where *name* is the GENPL name as listed above. Thus, if you want to list out all of the procedures and jobs that build a particular product, you can use UPDATE on GENPL and select the beginning and ending decks for that product. The following example demonstrates how to view all the generation procedures and jobs associated with the C compiler:

```
UPDATE,I=0,P=GENPL,N=0,C=0,Q='BGN-C.END-C',S.
```

In the example above, the $SR dataset now contains the C information. A similar UPDATE with any GENPL name will produce the same information for its corresponding product.

Each product section with GENPL is organized as follows:

BGN-*name*
name-ID common deck††
name-OWN common deck††
*name*PROCS common deck
all other common decks arranged alphabetically
END-*name*

The *name*PROCS common deck contains the generation procedures that are unique to that particular product.

† Products that are released asynchronously from COS.
†† Common decks that are only present for products that are released asynchronously from COS.

All other common decks within a product section contain generation jobs for that particular product. The names of these decks all begin with GEN. The MSC section also contains some miscellaneous GENPL common decks.

1.1.2 CONFIGURATION OF GENPL

GENPL is itself a PL. Modifications can be applied to it just like any other PL. You will have to make some modifications to GENPL, just as they will have to modify COSPL and IOPPL in order to reflect your particular hardware. The following list shows the decks that you may need to configure:

- ACCOUNT — This common deck contains an ACCOUNT statement that is used by all generation jobs. The ACCOUNT statement in the released PL will probably be invalid at your site.

- ASSIGN — This common deck, used in conjunction with the $ASSIGN procedure, allows all datasets to be preassigned to specified logical devices. Types passed to the ASSIGN common deck by GENPL are: BINARY, LISTING, TEMP, and UBINARY.

- AUDIT — This common deck is called at the end of all generation jobs that do not spawn a successor job.

- BASEED†, BASEID†, and BASEOWN† — These common decks contain the permanent dataset ID, edition, and owner used for datasets accessed as a base for system generation. These common decks are used only if the SW-NOBAS feature is disabled. These common decks are empty in the released PL.

- CPUH† — This common deck contains the CPU type passed on the CPU parameter to all generation procedures. Valid types are: CRAY-1, CRAY-1M, CRAY-1S, CRAY-XMP (non-EMA), CRAY-X4 (EMA), CRAY-XEA, and CRAY-YMP. This common deck is empty in the released PL. CPU targeting then defaults to the host machine characteristics. This common deck must be set correctly to verify that correct binaries are generated. It cannot be used for full system cross generation because once a binary is created, it is used to generate subsequent binaries.

† Common decks that must contain the desired value, followed by the COS JCL continuation character.

- ERROR — This common deck is called after the EXIT statement in all the generation procedures. The ERROR common deck in the released PL contains a DUMPJOB and a REWIND of $BLD and $CPL.

- REPLACE — This common deck can be used to generate a new system to replace an old system. This common deck is empty in the released PL, so no replacing is done. If this common deck is set to REPLACE, followed by the COS JCL continuation character, all old system binaries are deleted before a new system binary is saved.

- JOBEND — This common deck is called at the end of all generation jobs. The JOBEND common deck in the released PL contains two EXIT statements with comments on the termination of the job.

- ED†, ID†, and OWN† — Contains the permanent dataset ID, edition, and owner that are used for most datasets accessed and created by the generation jobs.

- IOSDIR† — This common deck contains the name of the directory on the IOS expander disk pack that is used by the GENDSDK job and DSDISK procedure. The IOSDIR common deck in the released PL contains a call to the ID common deck.

- IOSTAPE† — This common deck contains the name of the IOS expander tape that is used by the GENDSTP job and DSTAPE procedure. The IOSTAPE common deck in the released PL contains a call to the ID common deck.

- *name*-ID† and *name*-OWN† — These common decks contain the permanent dataset ID and owner used to access the necessary datasets to generate for the products released asynchronously from COS. Valid values for *name* are AUT, C, CFT, C77, IOS, MIG, PSC, and SL.

Use the UPDATE utility to modify these decks to fit your needs (see the UPDATE Reference Manual, publication SR-0013, for information on modifying decks).

There are certain conditional executional features you can control through common decks. You can use JCL control registers as switches to enable particular features. The following are the features you may want to modify:

- SW-COS24 — Generate COS with 24-bit addressing.

---

† Common decks that must contain the desired value, followed by the COS JCL continuation character.

- SW-DIANL Do not create and save diagnostic listings.
- SW-DIAX Generate diagnostics for CRAY X-MP systems. CRAY-1 systems must disable this feature or change the CPUH common deck to properly generate diagnostics.
- SW-GO Continue generation and ignore errors. An abort still occurs at job termination. The GENCOS job is the only job that currently uses this feature.
- SW-LIST Create and save listings.
- SW-NOBAS Generate a system using the binaries in the system directory as a base.
- SW-SPAWN Each basic utility generation job, GENPROC through GENSKOL, spawns its successor.

These features are all enabled by default. You can disable any of the features above by deleting the second line of the common deck with the UPDATE utility (see the UPDATE Reference Manual, publication SR-0013, for information on modifying a deck). The following example shows how to disable the SW-LIST feature:

```
*DELETE SW-LIST.2
```

1.1.3 PROCLIB

The heart of GENPL is the Procedure Library (PROCLIB). PROCLIB is the dataset created by the Control Statement Processor (CSP) in the GENPROC job. PROCLIB contains all of the procedures, in executable form, in GENPL. Refer to the COS Reference Manual, publication SR-0011, for information on the creation of procedure libraries.

The GENPL procedures fall into one of three classes depending on the first character of the procedure name. These three classes and their descriptions follow:

Class	Description
G*name*	Generate *name* routine
$xxxx	Low-level procedure
other	Miscellaneous procedure

1.2 GENPL GENERATION JOBS

The generation jobs are divided into categories that separate the jobs according to the software they generate. The following is a list of the categories of generation jobs:

- Basic utility generation jobs
- Supporting product generation jobs
- Operating system generation jobs
- Miscellaneous jobs

Each generation job is contained within its own common deck found in GENPL, and each job name is the same as the common deck name. The individual routines in each generation job are generated in order of decreasing memory usage. This allows each job to make the most efficient use of its allocated memory.

Several dataset naming conventions are used within the generation jobs. Local datasets, which are not intended to be made permanent, are prefixed with a $. Intermediate versions of various routines, that are not intended to be used outside of the generation process, are prefixed with a %. Listing datasets are suffixed with an LS, but not all datasets ending with an LS are listing datasets.

An important aspect of the generation jobs is that routines generated by previous jobs (and routines generated earlier inside a given job) may be used in the generation of subsequent routines. In this way, configuration changes and other interactions propagate through the software. Figure 1-2 shows the structure of a typical generation job.

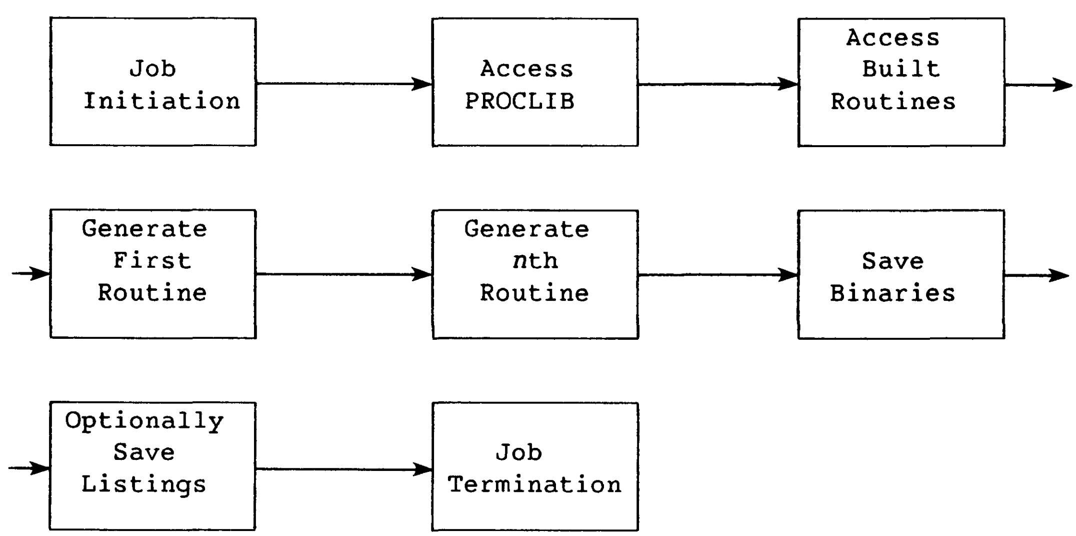

Figure 1-2. Generation Job Structure

1.2.1 BASIC UTILITY GENERATION JOBS

The basic utility generation job series builds the fundamental components of the system. This includes PROCLIB, the system definition files, the libraries, the compilers and assemblers, and other basic utilities (such as UPDATE, BUILD, and SEGLDR).

The basic utility generation job series builds most of its components twice so that all modification and configuration changes are fully propagated. Due to the sequential nature of these jobs, they are set up to automatically spawn the next job in the series. See the Release Notice for specific information on how to submit these jobs.

The following list shows the individual jobs in the basic utility generation series, along with their descriptions. They are listed in the order in which they are spawned.

Jobs	Description
GENPROC	Generate PROCLIB and save JSYSDIR
GENTCAL	Generate temporary CAL, $SYSDEF and $UTLDEF
GENTUTL	Generate temporary utilities (UPDATE, SEGLDR and BUILD)
GENTLIB	Generate temporary libraries ($UTLIB, $ARLIB, $FTLIB, $IOLIB, $SCILIB and $SYSLIB)
GENTPSC	Generate temporary PASCAL and $PSCLIB
GENCAL	Generate CAL, $SYSDEF and $UTLDEF
GENTC77	Generate temporary CFT77
GENBSCC	Generate Bootstrap C 3.1
GENCC	Generate C 3.1
GENUTL	Generate utilities (UPDATE, SEGLDR and BUILD)
GENLIB	Generate libraries ($UTLIB, $ARLIB, $FTLIB, $IOLIB, $SCILIB and $SYSLIB)
DELTEMP	Delete temporary datasets
GENSKOL	Generate SKOL, SKOLREF and SKOLTXT
GENCOSL	Generate COS libraries ($RDMLIB, $ADMLIB and $PDSLIB)

1.2.2 SUPPORTING PRODUCT GENERATION JOBS

The supporting product generation jobs build the remaining software products. These jobs include the supporting tools and utilities. These jobs use the routines built by the basic utility generation jobs. Most of these jobs can run concurrently with each other. See the Release Notice for specific interjob dependencies and information on submitting these jobs.

The following is a list of the supporting product generation jobs, along with their descriptions:

Job	Description
GENARCH	Generate archiving utilities
GENAUT	Generate Autotasking utilities
GENCFT	Generate CFT
GENCOSU†	Generate COS utilities
GENCSM	Generate CSIM
GENDBUG	Generate debuggers
GENDIA	Generate on-line diagnostics
GENGOS	Generate GOS utilities
GENMIG	Generate COS migration tools
GENMULT	Generate multitasking libraries
GENMUL1	
GENMUL2	
GENNCC	Generate C 4.0 user compiler
GENOCAL	Generate OLDCAL, $SYSTXT, COSTXT, $UTLTXT, and, optionally, stack, multitasking and SUPERLINK versions of $SYSTXT and $UTLTXT
GENPRD1†	Generate products (part 1)
GENPRD2†	Generate products (part 2)
GENRDM	Generate RDM utilities
GENSLMU	Generate SUPERLINK multitasking libraries
GENSLM1	
GENSLM2	
GENSLST	Generate SUPERLINK stack libraries
GENSLS1	
GENSLS2	
GENSM45	Generate COS Table Descriptions Manual
GENSTCK	Generate stack libraries
GENSTC1	
GENSTC2	
GENTAPE	Generate tape utilities
GENTOOL	Generate Software Tools and LD2

1.2.3 OPERATING SYSTEM GENERATION JOBS

The operating system generation jobs build the operating system components for the Cray mainframe and for the I/O Subsystem. These jobs use the routines built by the basic utility generation jobs and the basic supporting product generation jobs (prefixed with a † in the list in the previous section). The operating system generation jobs may run concurrently with each other. See the Release Notice for specific information on submitting these jobs.

† Jobs that generate the basic supporting products. The rest of the jobs are optional and need only be run if you want to make use of a job's corresponding feature.

The following list shows the operating system generation jobs, along with their descriptions:

Generation Job	Description
GENCOS	Generate COS software
GENIOS	Generate IOS software

The GENIOS job is optional for non-IOS sites.

1.2.4 MISCELLANEOUS JOBS

Several miscellaneous jobs are provided in GENPL. These are jobs that build the system directory and generate deadstart media. See the Release Notice for specific information on submitting these jobs.

The following is a list of these miscellaneous jobs, along with their descriptions:

Job	Description
GENDSDK	Generate deadstart expander disk
GENDSTP	Generate deadstart expander tape
JSYSDIR	System directory job

1.3 GENPL GENERATION PROCEDURES

There are generation procedures for all parts of the COS software, including many low-level procedures that are invoked within the various procedures. The generation procedures listed throughout this section are available to the system analyst. All generation procedures reside in PROCLIB and are in executable form. The following is a list of all the generation procedures, along with the PL name containing the routine's source, and the common deck in GENPL where each procedure is found, and the generation job in GENPL where each procedure is called. Subsections 1.3.2 through 1.3.15 are arranged by GENPL common deck and describe these procedures in more detail. To find more information on a particular procedure, find the procedure in the following list, and then look it up in the GENPL common deck section.

Procedure	PL Name	GENPL Common Deck	GENPL Job
G$ADMLIB	COSUTPL	COSPROCS	GENCOSL
G$ARLIB	ARLIBPL	LIBPROCS	GENTLIB,GENLIB†
G$CLIB	CPL	CPROCS	GENCC,GENNCC
G$DBHELP	SIDPL	PRDPROCS	GENPRD2
G$DBMSGS	DBGPL	PRDPROCS	GENDBUG
G$DIAGLB	DIAGPL,††	DIAPROCS	GENDIA
G$FTLIB	FTLIBPL	LIBPROCS	GENTLIB,GENLIB†
G$IOLIB	IOLIBPL	LIBPROCS	GENTLIB,GENLIB†
G$LEXLB	TOOLPL	PRDPROCS	GENTOOL
G$PDDERR	TOOLPL	PRDPROCS	GENTOOL
G$PDSLIB	PDSLBPL	COSPROCS	GENCOSL
G$PERFLB	UTILPL	PRDPROCS	GENPRD2
G$PSCLIB	PASCLPL	PSCPROCS	GENPSC,GENMUL1, GENSLM1,GENTPSC
G$RATDEF	TOOLPL	PRDPROCS	GENTOOL
G$RATLIB	TOOLPL	PRDPROCS	GENTOOL
G$RATXVS	TOOLPL	PRDPROCS	GENTOOL
G$RDMLIB	COSUTPL,COSPL	COSPROCS	GENCOSL
G$SCILIB	SCILBPL	LIBPROCS	GENTLIB,GENLIB†
G$SID	SIDPL	PRDPROCS	GENPRD2,GENMULT
G$SLLIB	SLLIBPL	SLPROCS	GENSLM2,GENSLS2
G$SYSDEF	SYSDFPL,COSPL	PRDPROCS	GENTCAL,GENCAL†
G$SYSLIB	SYSLBPL	LIBPROCS	GENTLIB,GENLIB†
G$SYSTXT	SYSDFPL,COSPL	PRDPROCS	GENOCAL
G$TDGTOC	TOOLPL	PRDPROCS	GENTOOL
G$UTLDEF	COSPL,SYSDFPL	PRDPROCS	GENTCAL,GENCAL†
G$UTLIB	UTLIBPL,DBGPL	LIBPROCS	GENTLIB,GENLIB†
G$UTLTXT	COSPL,SYSDFPL	PRDPROCS	GENOCAL
G$WTOOLS	TOOLPL	PRDPROCS	GENTOOL
G$YYPLB	TOOLPL	PRDPROCS	GENTOOL
GAACDEF	SLLIBPL	SLPROCS	GENSLS2
GACCOUNT	COSUTPL	COSPROCS	GENCOSU
GACCRDM	COSUTPL	COSPROCS	GENRDM
GACCTDEF	COSUTPL	COSPROCS	GENCOSU
GADSTAPE	LDRPL	PRDPROCS	GENPRD1
GALTBCD	COSUTPL,PDSLBPL	COSPROCS	GENARCH
GAPML	CALPL	PRDPROCS	GENPRD1
GAUDIT	COSUTPL	COSPROCS	GENCOSU
GAUDPL	UPDPL	PRDPROCS	GENPRD2
GBACKUP	COSUTPL	COSPROCS	GENARCH
GBIND	LDRPL	PRDPROCS	GENPRD1

† Procedures that are also called from the GENMULT, GENSTCK, GENSLMU, and GENSLST generation job series.
†† Hardware specific PL (XMPPL or CRAY1PL).

Procedure	PL Name	GENPL Common Deck	GENPL Job
GBINDGOS	COSUTPL	COSPROCS	GENGOS
GBLOCK	TOOLPL	PRDPROCS	GENPRD1
GBLOKGOS	COSUTPL	COSPROCS	GENGOS
GBMXTAP	DIAGPL	DIAPROCS	GENDIA
GBUILD	LDRPL	PRDPROCS	GENTUTL,GENUTL
GBUPIO	COSUTPL	COSPROCS	GENARCH
GBVCEDIT	COSUTPL	COSPROCS	GENARCH
GCAL	CALPL	PRDPROCS	GENTCAL,GENCAL
GCC	CPL	CPROCS	GENCC,GENNCC
GCFT	CFTPL	CFTPROCS	GENCFT
GCFT77	CFT77PL	C77PROCS	GENC77,GENTC77
GCHARGES	COSUTPL	COSPROCS	GENCOSU
GCHNGGOS	COSUTPL	COSPROCS	GENGOS
GCLEANUP	COSUTPL	COSPROCS	GENARCH
GCLEARIO	DIAGPL	DIAPROCS	GENDIA
GCLUPIO	COSUTPL	COSPROCS	GENARCH
GCOMPARE	UTILPL	PRDPROCS	GENPRD2
GCONF	DIAGPL	DIAPROCS	GENDIA
GCONNECT	UTILPL	SLPROCS	GENSLS2
GCOPYD	UTILPL	PRDPROCS	GENPRD2,GENSLS2
GCOPYF	UTILPL	PRDPROCS	GENPRD2,GENSLS2
GCOPYNF	UTILPL	PRDPROCS	GENPRD2
GCOPYR	UTILPL	PRDPROCS	GENPRD2,GENSLS2
GCOPYU	UTILPL	PRDPROCS	GENPRD2
GCOS	COSPL	COSPROCS	GENCOS
GCOSDEF	COSPL,SYSDFPL	COSPROCS	GENCOS
GCOSJCL	MIGPL	MIGPROCS	GENMIG
GCOSPAR	MIGPL	MIGPROCS	GENMIG
GCOSTXT	COSPL,SYSDFPL	PRDPROCS	GENOCAL
GCPP	CPL	CPROCS	GENCC,GENNCC
GCSIM	CSIMPL	CSMPROCS	GENCSM
GCSP	COSPL	COSPROCS	GENCOS
GDDA	DBGPL	PRDPROCS	GENDBUG
GDDC	COSPL	COSPROCS	GENCOS
GDDTEST	DIAGPL	DIAPROCS	GENDIA
GDEBUG	DBGPL	PRDPROCS	GENDBUG
GDEC	COSPL	COSPROCS	GENCOS
GDELAY	DIAGPL	DIAPROCS	GENDIA
GDIA	COSPL	COSPROCS	GENCOS
GDIAGLB	DIAGPL	DIAPROCS	GENDIA
GDONUT	DIAGPL	DIAPROCS	GENDIA
GDOWNCPU	DIAGPL,XMPPL	DIAPROCS	GENDIA
GDQM	COSPL	COSPROCS	GENCOS

† Procedures that are also called from the GENMULT, GENSTCK, GENSLMU, and GENSLST generation job series.

Procedure	PL Name	GENPL Common Deck	GENPL Job
GDRD	DBGPL	PRDPROCS	GENDBUG
GDSDIAG	DIAGPL	DIAPROCS	GENDIA
GDSDUMP	UTILPL	PRDPROCS	GENPRD2
GDUMP	COSUTPL	COSPROCS	GENCOSU
GDUMPGOS	COSUTPL	COSPROCS	GENGOS
GEXEC	COSPL	COSPROCS	GENCOS
GEXP	COSPL	COSPROCS	GENCOS
GEXTRACT	COSUTPL	COSPROCS	GENCOSU
GFDUMP	COSUTPL	COSPROCS	GENCOSU
GFLODUMP	UTILPL	PRDPROCS	GENPRD2
GFMP	FMPPL	AUTPROCS	GENAUT
GFPP	FPPPL	AUTPROCS	GENAUT
GFRLS	SLLIBPL	SLPROCS	GENSLS2
GFSELECT	SLLIBPL	SLPROCS	GENSLS2
GFTREF	TOOLPL	PRDPROCS	GENPRD1
GFVD	COSPL	COSPROCS	GENCOS
GGENBCD	COSUTPL,PDSLBPL	COSPROCS	GENARCH
GGENMCD	COSUTPL,PDSLBPL	COSPROCS	GENARCH
GHERG	DIAGPL	DIAPROCS	GENDIA
GINITT	COSUTPL,COSPL	COSPROCS	GENTAPE
GIOS	IOPPL	IOSPROCS	GENIOS
GIQM	COSPL	COSPROCS	GENCOS
GISP	UTILPL	SLPROCS	GENSLST
GITEMIZE	UTILPL	PRDPROCS	GENPRD2
GJCM	COSPL	COSPROCS	GENCOS
GJCSDEF	COSUTPL	COSPROCS	GENCOSU
GJOBSIZE	MIGPL	MIGPROCS	GENMIG
GJSH	COSPL	COSPROCS	GENCOS
GLD2	LDRPL	PRDPROCS	GENTOOL
GLDR	LDRPL	PRDPROCS	GENPRD1,GENSLMU,GENSLST
GLOADCAT	COSUTPL	COSPROCS	GENARCH
GLOADGOS	COSUTPL	COSPROCS	GENGOS
GMAINT	†	DIAPROCS	GENDIA
GMANAGE	COSUTPL	COSPROCS	GENARCH
GMENULIB	DIAGPL	DIAPROCS	GENDIA
GMEP	COSPL	COSPROCS	GENCOS
GMIGRATE	COSUTPL	COSPROCS	GENARCH
GMODECKS	UPDPL	PRDPROCS	GENPRD2
GMODSEQ	TOOLPL	PRDPROCS	GENPRD1
GMODSET	TOOLPL	PRDPROCS	GENPRD1
GMSG	COSPL	COSPROCS	GENCOS
GMTDUMP	UTILPL	PRDPROCS	GENPRD2
GNOTE	TOOLPL	PRDPROCS GENPL	GENPRD1

† Hardware specific PL (XMPPL or CRAY1PL)

Procedure	PL Name	Common Deck	GENPL Job
GNUPDATE	NUPDPL	PROPROCS	GENPRD1
GOFFCONF	DIAGPL	DIAPROCS	GENDIA
GOLCFDT	DIAGPL	DIAPROCS	GENDIA
GOLDCAL	CALPL	PRDPROCS	GENOCAL
GOLDMON	DIAGPL	DIAPROCS	GENDIA
GOLNET	DIAGPL	DIAPROCS	GENDIA
GPASCAL	PASCLPL	PSCPROCS	GENPSC,GENTPSC
GPASSWRD	COSUTPL	COSPROCS	GENRDM
GPDM	COSPL	COSPROCS	GENCOS
GPDMCAT	COSUTPL,PDSLBPL	COSPROCS	GENARCH
GPDSDUMP	COSUTPL	COSPROCS	GENCOSU,GENSLS2
GPDSLOAD	COSUTPL	COSPROCS	GENCOSU,GENSLS2
GPERFMON	TOOLPL	PRDPROCS	GENPRD1
GPREMULT	PMULTPL	PRDPROCS	GENPRD2
GPROCLIB	GENPL	MSCPROCS	GENPROC
GPROFILE	COSUTPL	COSPROCS	GENRDM
GPRVDEF	COSUTPL	COSPROCS	GENCOSU
GQSTSUI	TOOLPL	PRDPROCS	GENTOOL
GQUERY	UTILPL	PRDPROCS	GENPRD2
GRDACC	COSUTPL	COSPROCS	GENRDM
GRDAGET	COSUTPL	COSPROCS	GENRDM
GRDAPUT	COSUTPL	COSPROCS	GENRDM
GRDAUDIT	COSUTPL	COSPROCS	GENRDM
GRDEDIT	COSUTPL	COSPROCS	GENRDM
GRDGEN	COSUTPL	COSPROCS	GENRDM
GRDM	COSPL	COSPROCS	GENCOS
GRDMERGE	COSUTPL,COSPL	COSPROCS	GENRDM
GRDNRD	COSUTPL,COSPL	COSPROCS	GENRDM
GRDQSC	COSUTPL,COSPL	COSPROCS	GENRDM
GRDSCAN	COSUTPL,COSPL	COSPROCS	GENRDM
GRDVAL	COSUTPL,PDSLBPL	COSPROCS	GENRDM
GRECALL	COSUTPL	COSPROCS	GENARCH
GRECIO	COSUTPL	COSPROCS	GENARCH
GRELEASE	SLLIBPL	SLPROCS	GENSLS2
GRELOAD	COSUTPL	COSPROCS	GENARCH
GRESTORE	COSUTPL	COSPROCS	GENARCH
GRETIRE	COSUTPL	COSPROCS	GENARCH
GREWIND	SLLIBPL	SLPROCS	GENSLS2
GROUTE	COSUTPL	COSPROCS	GENCOSU
GSCP	COSPL	COSPROCS	GENCOS
GSEGLDR	LDRPL	PRDPROCS	GENTUTL,GENUTL,GENSLMU GENSLST
GSEGRLS	LDRPL	PRDPROCS	GENPRD1
GSETOWN	COSUTPL	COSPROCS	GENCOSU
GSKIPD	UTILPL	PRDPROCS	GENPRD2,GENSLS2
GSKIPF	UTILPL	PRDPROCS	GENPRD2,GENSLS2
GSKIPR	UTILPL	PRDPROCS	GENPRD2,GENSLS2
GSKIPU	UTILPL	PRDPROCS	GENPRD2
GSKOL	SKOLPL	PRDPROCS	GENSKOL

Procedure	PL Name	Common Deck	GENPL Job
GSKOLREF	SKOLPL	PRDPROCS	GENSKOL
GSKOLTXT	SKOLPL	PRDPROCS	GENSKOL
GSLSUB	SLLIBPL	SLPROCS	GENSLM2
GSLT	COSPL	COSPROCS	GENCOS
GSORT	UTILPL	PRDPROCS	GENPRD2
GSPAWN	TOOLPL	PRDPROCS	GENPRD1
GSPM	COSPL	COSPROCS	GENCOS
GSPY	UTILPL	PRDPROCS	GENPRD2
GSSAF	COSUTPL	COSPROCS	GENCOSU
GSTARTUP	COSPL	COSPROCS	GENCOS
GSTATGOS	COSUTPL	COSPROCS	GENGOS
GSTATMCD	COSUTPL,PDSLBPL	COSPROCS	GENARCH
GSTATS	COSUTPL	COSPROCS	GENCOSU
GSTEP	TOOLPL	PRDPROCS	GENPRD1
GSTG	COSPL	COSPROCS	GENCOS
GSTOPGOS	COSUTPL	COSPROCS	GENGOS
GSTP	COSPL	COSPROCS	GENCOS
GSTPCOM	COSPL	COSPROCS	GENCOS
GSTPTAB	COSPL	COSPROCS	GENCOS
GSYSREF	TOOLPL	PRDPROCS	GENPRD1
GTARGET	UTILPL	PRDPROCS	GENPRD2
GTASKS	COSPL	COSPROCS	GENCOS
GTDI	COSUTPL,COSPL	COSPROCS	GENTAPE
GTDUMP	COSUTPL,COSPL	COSPROCS	GENTAPE
GTEDI	TEDIPL	PRDPROCS	GENPRD2
GTG	COSUTPL	COSPROCS	GENTAPE
GTQM	COSPL	COSPROCS	GENCOS
GUNB	UTILPL	PRDPROCS	GENPRD1
GUNBLOCK	TOOLPL	PRDPROCS	GENPRD1
GUPDATE	UPDPL	PRDPROCS	GENTUTL,GENUTL
GUPIC	UPDPL	PRDPROCS	GENPRD2
GVALBCD	COSUTPL,PDSLBPL	COSPROCS	GENARCH
GVOLMAP	COSUTPL	COSPROCS	GENTAPE
GWRITEDS	UTILPL	PRDPROCS	GENPRD2

1.3.1 PROCEDURE FORMAT AND PARAMETERS

Each entity (compiler, library, utility, tool, etc.) in the system directory (SDR) has its own generation procedure. Each generation procedure is named after the routine that it generates. The name of the procedure consists of a G, followed by the routine name. For example, the procedure that generates the arithmetic library ($ARLIB) is named G$ARLIB. These procedures are defined in common decks throughout GENPL.

Except when noted, all generation procedures have the following parameters:

Parameter	Default	Purpose
ALLOC	STATIC	Selects the type of memory allocation for Fortran library routines. This parameter is available on all library generation procedures and procedures used to generate stack and multitasking versions of software. ALLOC parameter values are STATIC and STACK.
B	*name*1	Selects a binary dataset.
CL	*name*3a	Selects an alternate PL dataset. This parameter is available only on those procedures that require it.
CPU	(none)	Selects the CPU targeting for the binary dataset. This parameter can be changed with the CPUH common deck (see subsection 1.1.2).
DEF	(none)	Selects symbols to be defined for UPDATE. These symbols are used by UPDATE when processing *IF directives. If multiple symbols are desired, they must be enclosed in parentheses: "DEF=(STACK:MULTI)".
I	0	Selects an UPDATE input dataset.
L	*name*2	Selects a listing, cross-reference, and BUILD/SEGLDR output dataset.
N	0	Selects a new UPDATE PL dataset.
P	*name*3b	Selects an UPDATE PL dataset
TYPE	BINARY	Selects the type of binary dataset being created. This parameter is available only on the generation procedures that are called more than once by the generation jobs. It can be used in the ASSIGN common deck to determine the type of ASSIGN to perform on a binary dataset. The parameter values are BINARY and TEMP.
U	0	Selects an UPDATE listing dataset.

Note that the default binary, listing and PL dataset names (*name*1, *name*2, *name*3a, and *name*3b) vary from procedure to procedure. For example, the prototype (definition) statement for the $ARLIB generation procedure is as follows:

```
G$ARLIB(ALLOC=STATIC:STATIC,B=$ARLIB:$ARLIB,CPU=:,DEF=:,^
  I=0:0,L=ARLIBLS:ARLIBLS,P=ARLIBPL:ARLIBPL,N=0:0,^
  TYPE=BINARY:BINARY,U=0:0)
```

The default name for the binary dataset generated by this procedure is $ARLIB (*name*1). The default name for the listing dataset is ARLIBLS (*name*2), and the default name for the PL dataset name is ARLIBPL (*name*3b). There is no alternate PL dataset name (*name*3a) for this procedure. Figure 1-3 shows the control options and data flow for G$ARLIB, a typical generation procedure.

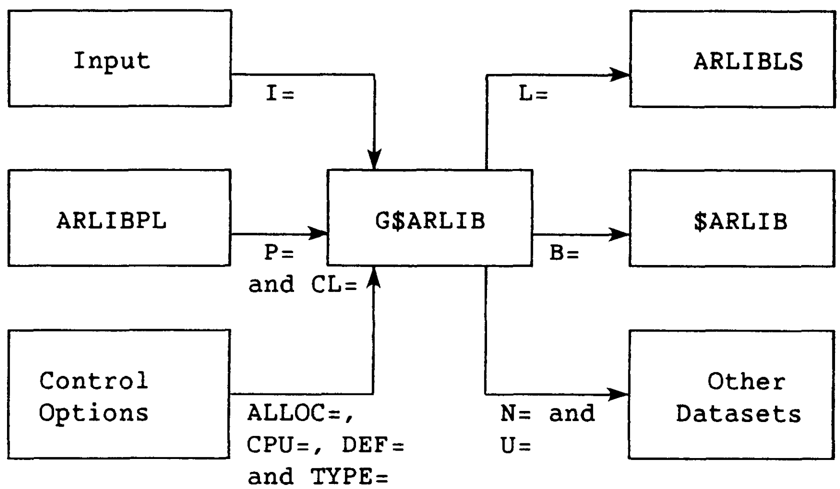

Figure 1-3. G$ARLIB Generation Procedure Schematic

All procedures assume that the input datasets (I, P, and CL) are local to the job. The procedures have the following dataset positioning conventions:

B	Released, written and rewound (unless B=0)
CL	Positioned by UPDATE (if used)
I	Read from current position (unless I=0), not rewound
L	Written at current position (unless L=0),· not rewound
P	Positioned by UPDATE (unless P=0)
N	Positioned by UPDATE (unless N=0)
U	Written at current position (unless U=0), not rewound

Any temporary datasets created by a procedure will be released before the procedure terminates and will have a name that is prefixed with a dollar sign ($).

1.3.2 AUTPROCS GENERATION PROCEDURES

The following are the Autotasking generation procedures defined in the common deck AUTPROCS in GENPL:

Procedure	PL Name	Binary	Listing
GFMP	FMPPL	FMP	FMPLS
GFPP	FPPPL	FPP	FPPLS

1.3.3 CPROCS GENERATION PROCEDURES

The following are the C compiler and library generation procedures defined in the common deck CPROCS in GENPL:

Procedure	PL Name	Binary	Listing
G$CLIB	CPL	$CLIB	CLIBLS
GCC	CPL	CC	CCLS
GCPP	CPL	CPP	CPPLS

1.3.4 CFTPROCS GENERATION PROCEDURES

The following are the CFT generation procedures defined in the common deck CFTPROCS in GENPL:

Procedure	PL Name	Binary	Listing
GCFT	CFTPL	CFT	CFTLS

1.3.5 C77PROCS GENERATION PROCEDURES

The following are the CFT77 generation procedures defined in the common deck C77PROCS in GENPL:

Procedure	PL Name	Binary	Listing
GCFT77	CFT77PL	CFT77	CFT77LS

The GCFT77 procedure also has a few special parameters to allow various types of compiler builds. These parameters are:

Name	Default	Purpose
DBGLVL	NORMAL	Selects the desired level of debugging on a debugger compiler. Valid types are NORMAL and FULL. This parameter is only meaningful when used with the DEBUG parameter.
DEBUG	(none)	Selects a debugger compiler build. This parameter cannot be equated.
NONSEG	NONSEG	Selects a nonsegmented compiler build. 1- and 2-Mword sites must set NONSEG to FALSE to generate the release software.

1.3.6 COSPROCS GENERATION PROCEDURES

The COSPROCS common deck contains the COS library and utility generation procedures in GENPL. The following is a listing of these procedures:

Procedure	PL Name	Binary	Listing
G$ADMLIB	COSUTPL	$ADMLIB	ADMLBLS
G$PDSLIB	PDSLBPL	$PDSLIB	PDSLBLS
G$RDMLIB†	COSUTPL,COSPL	$RDMLIB	RDMLBLS
GACCOUNT	COSUTPL	ACCOUNT	ACCTLS
GACCRDM	COSUTPL	ACCRDM	ACRDMLS
GACCTDEF	COSUTPL	ACCTDEF	ACTDFLS
GALTBCD†	COSUTPL,PDSLBPL	ALTBCD	ABCDLS
GAUDIT	COSUTPL	AUDIT	AUDITLS
GBACKUP	COSUTPL	BACKUP	BCKUPLS
GBINDGOS	COSUTPL	BINDGOS	BINDGLS
GBLOKGOS	COSUTPL	BLOKGOS	BLOKGLS
GBUPIO	COSUTPL	BUPIO	BUPIOLS
GBVCEDIT	COSUTPL	BVCEDIT	BVCEDLS
GCHARGES	COSUTPL	CHARGES	CHARGLS
GCHNGGOS	COSUTPL	CHNGGOS	CHNGGLS
GCLEANUP	COSUTPL	CLEANUP	CLEANLS
GCLUPIO	COSUTPL	CLUPIO	CLUPLS
GCOSDEF†,††	COSPL,SYSDFPL	COSDEF	CSDEFLS
GDDC	COSPL	DDC	DDCLS
GDUMP	COSUTPL	DUMP	DUMPLS
GDUMPGOS	COSUTPL	DUMPGOS	DUMPGLS
GEXTRACT	COSUTPL	EXTRACT	EXTRALS

† Procedure has a CL parameter
†† Procedure has a default DEF=CALV2

Procedure	PL Name	Binary	Listing
GFDUMP	COSUTPL	FDUMP	FDUMPLS
GGENBCD†	COSUTPL,PDSLBPL	GENBCD	GBCDLS
GGENMCD†	COSUTPL,PDSLBPL	GENMCD	GMCDLS
GINITT†	COSUTPL,COSPL	INITT	INITTLS
GJCSDEF	COSUTPL	JCSDEF	JCSDFLS
GLOADCAT	COSUTPL	LOADCAT	LOADCLS
GLOADGOS	COSUTPL	LOADGOS	LOADGLS
GMANAGE	COSUTPL	MANAGE	MANAGLS
GMIGRATE	COSUTPL	MIGRATE	MIGRLS
GPASSWRD	COSUTPL	PASSWRD	PASSWLS
GPDMCAT†	COSUTPL,PDSLBPL	PDMCAT	PDCATLS
GPDSDUMP	COSUTPL	PDSDUMP	PDUMPLS
GPDSLOAD	COSUTPL	PDSLOAD	PLOADLS
GPROFILE	COSUTPL	PROFILE	PROFLS
GPRVDEF	COSUTPL	PRVDEF	PRVDFLS
GRDACC	COSUTPL	RDACC	RDACCLS
GRDAGET	COSUTPL	RDAGET	RDAGTLS
GRDAPUT	COSUTPL	RDAPUT	RDAPTLS
GRDAUDIT	COSUTPL	RDAUDIT	RDAUDLS
GRDEDIT	COSUTPL	RDEDIT	RDEDTLS
GRDGEN	COSUTPL	RDGEN	RDGENLS
GRDMERGE†	COSUTPL,COSPL	RDMERGE	RDMRGLS
GRDNRD†	COSUTPL,COSPL	RDNRD	RDNRDLS
GRDQSC†	COSUTPL,COSPL	RDQSC	RDQSCLS
GRDSCAN†	COSUTPL,COSPL	RDSCAN	RDSCNLS
GRDVAL†	COSUTPL,PDSLBPL	RDVAL	RDVALLS
GRECALL	COSUTPL	RECALL	RECALLS
GRECIO	COSUTPL	RECIO	RECIOLS
GRELOAD	COSUTPL	RELOAD	RELODLS
GRESTORE	COSUTPL	RESTORE	RESTLS
GRETIRE	COSUTPL	RETIRE	RETIRLS
GROUTE	COSUTPL	ROUTE	ROUTELS
GSETOWN	COSUTPL	SETOWN	SETONLS
GSSAF	COSUTPL	SSAF	SSAFLS
GSTATGOS	COSUTPL	STATGOS	STATGLS
GSTATMCD†	COSUTPL,PDSLBPL	STATMCD	SMCDLS
GSTATS	COSUTPL	STATS	STATSLS
GSTOPGOS	COSUTPL	STOPGOS	STOPGLS
GTDI†	COSUTPL,COSPL	TDI	TDILS
GTDUMP†	COSUTPL,COSPL	TDUMP	TDUMPLS
GTG	COSUTPL	TG	TGLS
GVALBCD†	COSUTPL,PDSLBPL	VALBCD	VBCDLS
GVOLMAP	COSUTPL	VOLMAP	VMAPLS

---

† Procedure has a CL parameter
†† Procedure has a default DEF=CALV2

The GCOS procedure is also defined in COSPROCS. GCOS generates the major operating system components of COS. This procedure differs from the generation procedures listed above in that it produces multiple binary and listing datasets.

The GCOS procedure has the following parameters:

Name	Default	Purpose
CPU	(none)	Selects the CPU targeting for the binaries generated by this procedure. This parameter can be changed with the CPUH common deck, or the SW-COS24 feature (see subsection 1.1.2).
DEF	CALV2	Selects symbols to be defined for UPDATE. These symbols are used by UPDATE when processing *IF directives. If multiple symbols are desired, they must be enclosed in parentheses.
L	$OUT	Selects a SEGLDR relocatable load map output dataset.
LISTING	ON	Selects the creation of assembly listings for each COS task.
P	COSPL	Selects an UPDATE PL dataset.

The GCOS procedure produces the following binary datasets:

Name	Description
COS	COS binary
COSLIB	Relocatable COS routines in BUILD library format
EXECSYM	EXEC symbol table (for FDUMP)
STPSYM	STP symbol table (for FDUMP)

The GCOS procedure calls the following procedures, which can also be invoked separately:

Procedure	PL Name	Binary	Listing
GEXEC	COSPL	$BLD	EXECLS
GSTPTAB	COSPL	$BLD	STPTBLS
GSTPCOM	COSPL	$BLD	STPCMLS
GSTP	COSPL	$BLD	TSTPLS
GSCP	COSPL	$BLD	TSCPLS
GEXP	COSPL	$BLD	TEXPLS
GTQM	COSPL	$BLD	TTQMLS
GMSG	COSPL	$BLD	TMSGLS
GJSH	COSPL	$BLD	TJSHLS

Procedure	PL Name	Binary	Listing
GJCM	COSPL	$BLD	TJCMLS
GDQM	COSPL	$BLD	TDQMLS
GDEC	COSPL	$BLD	TDECLS
GPDM	COSPL	$BLD	TPDMLS
GMEP	COSPL	$BLD	TMEPLS
GSPM	COSPL	$BLD	TSPMLS
GSTG	COSPL	$BLD	TSTGLS
GFVD	COSPL	$BLD	TFVDLS
GIQM	COSPL	$BLD	TIQMLS
GDIA	COSPL	$BLD	TDIALS
GSLT	COSPL	$BLD	TSLTLS
GRDM	COSPL	$BLD	TRDMLS
GSTARTUP	COSPL	$BLD	STARTLS
GTASKS	COSPL	$BLD	(none)
GCSP	COSPL	$BLD	CSPLS

The GCOS procedure generates the following listings:

Name	Description
CSPLS	CSP
EXECLS	EXEC
STARTLS	STARTUP
STPCMLS	STP common routines
STPTBLS	STP tables
TDECLS	Task DEC
TDIALS	Task DIA
TDQMLS	Task DQM
TEXPLS	Task EXP
TFVDLS	Task FVD
TIQMLS	Task IQM
TJCMLS	Task JCM
TJSHLS	Task JSH
TMEPLS	Task MEP
TMSGLS	Task MSG
TPDMLS	Task PDM
TRDMLS	Task RDM
TSCPLS	Task SCP
TSLTLS	Task SLT
TSPMLS	Task SPM
TSTGLS	Task STG
TSTPLS	Task STP
TTQMLS	Task TQM

The GCOSREF procedure is also defined in COSPROCS. GCOSREF produces no binary dataset, only a COS cross-reference listing dataset, COSRFLS. This procedure must only be invoked while the listing datasets created by the GCOS procedure are still local to the job.

1.3.7 CSMPROCS GENERATION PROCEDURES

The following are the CSIM generation procedures defined in the common deck CSMPROCS in GENPL:

Procedure	PL Name	Binary	Listing
GCSIM	CSIMPL	CSIM	CSIMLS

1.3.8 DIAPROCS GENERATION PROCEDURES

These are the on-line diagnostic generation procedures defined in the common deck DIAPROCS in GENPL:

Procedure	PL Name	Binary	Listing
G$DIAGLB	DIAGPL	$DIAGLB	DIAGLS
GCLEARIO	DIAGPL	CLEARIO	CLEARLS
GDSDIAG	DIAGPL	DSDIAG,DSDIAGO	DSDIALS

DSDIAG is the DSDIAG binary dataset. DSDIAGO contains the accompanying overlays necessary to run DSDIAG.

The G$DIAGLB procedure also requires a hardware-specific PL; either XMPPL or CRAY1PL. The PL used by the procedure is determined by the CPU parameter. The default is XMPPL. CRAY-1 systems must change the CPU parameter with the CPUH common deck, or by disabling the SW-DIAX feature.

The GDOWNCPU procedure is also defined in DIAPROCS. This procedure is only for CRAY X-MP systems and uses DIAGPL and XMPPL to generate the down CPU diagnostics that it uses. Its listing dataset is DCPULS. CRAY-1 systems must change the CPU parameter with the CPUH common deck, or by disabling the SW-DIAX feature to avoid the GDOWNCPU procedure. The following is a list of the down CPU diagnostics: AHT, ARB, ARM, BRB, CMP, CMX, GTH, IBZ, MIT, SFA, SFM, SFR, SIS, SR3, SRA, SRB, SRL, SRS, STAN, SVC, TRB, VPP, VRA, VRL, VRN, VRR, VRS, VRX.

The GOFFCONF procedure is also defined in DIAPROCS. This procedure is for CRAY X-MP systems only and uses DIAGPL to generate the off-line confidence diagnostics. Its listing dataset is OCONFLS. The following is a list of the off-line confidence diagnostics: OFFCFPT, OFFCM, OFFCRIT, OFFCSVC, OFFIBUF.

All on-line diagnostic software is Cray Proprietary. By default, no listing datasets are produced by any of the on-line diagnostic generation procedures. To generate on-line diagnostic listing datasets, the SW-DIANL feature must be disabled.

Most of the on-line diagnostic routines reside in the $DIAGLB dataset. The generation procedures for the individual routines that make up $DIAGLB are also defined in the common deck DIAPROCS. The G$DIAGLB procedure invokes the corresponding generation procedure to build each diagnostic. Each of these procedures use DEF parameters that are appropriate for COS sites based on the CPU parameter. The default is for CRAY X-MP systems. CRAY-1 systems must change the CPU parameter to generate the maintenance diagnostics with the CPUH common deck, or by disabling the SW-DIAX feature.

The following are the on-line diagnostic generation procedures that are invoked by G$DIAGLB:

Procedure	PL Name	Binary
GBMXTAP	DIAGPL	BMXTAP
GCONF	DIAGPL	Confidence diagnostics
GDDTEST	DIAGPL	DDTEST
GDELAY	DIAGPL	DELAY
GDIAGLB	DIAGPL	DIAGLB
GDONUT	DIAGPL	DONUT
GHERG	DIAGPL	HERG
GMAINT	XMPPL or CRAY1PL	Maintenance diagnostics
GMENULIB	DIAGPL	MENULIB
GOLCFDT	DIAGPL	OLCFDT
GOLDMON	DIAGPL	OLDMON
GOLNET	DIAGPL	OLNET

The confidence diagnostics referred to in the preceding list are: OLCFPT, OLCM, OLCRIT, OLCSVC, OLIBUF.

The maintenance diagnostics referred to in the preceding list are: AHT, ARB, ARM, BRB, MIT, SFA, SFM, SFR, SIS, SR3, SRA, SRB, SRL, SRS, STAN, SVC, TRB, VRA, VRL, VRN, VRR, VRS.

CRAY X-MP specific maintenance diagnostics are: CMP, CMX, GTH, IBZ, VPP, VRX.

CRAY-1 specific maintenance diagnostics are CMD and VPOP.

1.3.9 IOSPROCS GENERATION PROCEDURE

The GIOS procedure is the only generation procedure in IOSPROCS. GIOS generates the major operating system components of the IOS. This procedure differs from other generation procedures in that it produces multiple binary and listing datasets.

The GIOS procedure has the following parameters:

Name	Default	Purpose
CPU	(none)	Selects CPU targeting for the binaries generated by this procedure. This parameter can be changed by the CPUH common deck (see subsection 1.1.2).
DEF	(none)	Selects symbols to be defined for UPDATE. These symbols are used by UPDATE when processing *IF directives. If multiple symbols are desired, they must be enclosed in parentheses.
L	$OUT	Selects a BUILD output dataset.
LISTING	ON	Selects the creation of assembly listings for each IOS overlay.
P	IOPPL	Selects an UPDATE PL dataset.

The GIOS procedure produces the following binary datasets:

Name	Description
$APTEXT	IOS system text for APML modules
$DISKLD	80 Mbyte disk boot routine
$DUMP	Printer dump routine
$IOSDEF	IOS system text for CAL modules
$KERNEL	IOS Kernel routine
$OVERLY	IOS overlays
$TAPELD	IOS Peripheral Expander tape boot routine
$VMEDMP	VME dump routine
$VMELD	VME boot routine
IOSLIB	Relocatable IOS routines in BUILD library format

The GIOS procedure generates the following listings:

Name	Description
APTXTLS	$APTEXT
IBMXOLS	Block Mux channel overlays
ICONCLS	Concentrator overlays
ICOVLLS	CAL overlays
IDIAGLS	Diagnostic overlays
IDISKLS	Diskload
IDSKOLS	Disk driving overlays
IDUMPLS	Dump
IEXPOLS	Expander device overlays

Name	Description
IFEILS	Front-end interface overlays
IFILELS	Mini-editor and file system overlays
IHSXLS	HSX channel overlays
IKERNLS	Kernel
IKRNOLS	Kernel overlays
ILOADLS	Tapeload
INSCOLS	Network Systems Corporation (NSC) driver overlays
INTERLS	Interactive overlays
IOSDFLS	$IOSDEF
IOSRFLS	IOS cross-reference
ISDMPLS	System dump and I/O Processor restart overlays
ISTATLS	Station overlays
ITAPELS	Block Mux tape handling overlays
IUCHNLS	User channel driver overlays
IUNIXLS	UNICOS support overlays
IVMDMLS	VME dump
IVMLDLS	VME boot
IVMELS	VME overlays

1.3.10 LIBPROCS GENERATION PROCEDURES

The following are the library generation procedures defined in the common deck LIBPROCS in GENPL.

Procedure	PL Name	Binary	Listing
G$ARLIB	ARLIBPL	$ARLIB	ARLIBLS
G$FTLIB	FTLIBPL	$FTLIB	FTLIBLS
G$IOLIB	IOLIBPL	$IOLI	BIOLIBLS
G$SCILIB	SCILBPL	$SCILIB	SCILBLS
G$SYSLIB	SYSLBPL	$SYSLIB	SYSLBLS

All of the LIBPROCS generation procedures have ALLOC and TYPE parameters.

The G$UTLIB procedure is also defined in LIBPROCS. This procedure differs from the generation procedures listed above in that it uses multiple PLs to create the $UTLIB binary file and UTLIBLS listing dataset. The following PLs are used in G$UTLIB: DBGPL, FDCPL, FLOWPL, HPMGRPL, MISCPL, MULTIPL, TARGPL, TBMGRPL. To accommodate these multiple PLs, the G$UTLIB procedure has unique DEF, I, P, N, and U parameters for each PL. The parameter name consists of the original parameter (DEF, I, P, N, or U) followed by the PL designator as follows:

PL Designator	PL Name
DBG	DBGPL
FDC	FDCPL
FLOW	FLOWPL
HPMG	HPMGRPL
MISC	MISCPL
MULT	MULTIPL
TARG	TARGPL
TBMG	TBMGRPL

For example, the MISCPL parameters are DEFMISC, IMISC, PMISC, NMISC, and UMISC. The DEFDBG parameter has a default value of (COS:XMP).

1.3.11 MIGPROCS GENERATION PROCEDURES

The following are the COS migration tools generation procedures defined in the common deck MIGPROCS in GENPL:

Procedure	PL Name	Binary	Listing
GCOSJCL	MIGPL	COSJCL	COSJCLS
GCOSPAR	MIGPL	COSPAR	COSPRLS
GJOBSIZE	MIGPL	JOBSIZE	JOBSZLS

1.3.12 MSCPROCS GENERATION PROCEDURES

The following is the miscellaneous generation procedure defined in the common deck MSCPROCS in GENPL:

Procedure	PL Name	Binary	Listing
GPROCLIB	GENPL	PROCLIB	PRCLBLS

1.3.13 PSCPROCS GENERATION PROCEDURES

The following are the PASCAL compiler and library generation procedures defined in the common deck PSCPROCS in GENPL:

Procedure	PL Name	Binary	Listing
G$PSCLIB †,††	PASCLPL	$PSCLIB	PSCLBLS
GPASCAL ††	PASCLPL	PASCAL	PASCLLS

† Procedure has an ALLOC parameter
†† Procedure has a TYPE parameter

1.3.14 PRDPROCS GENERATION PROCEDURES

These are the product generation procedures defined in the common deck PRDPROCS in GENPL:

Procedure	PL Name	Binary	Listing
G$DBHELP	SIDPL	$DBHELP	DBHLPLS
G$DBMSGS [4]	DBGPL	$DBMSGS	DBMSGLS
G$LEXLB	TOOLPL	$LEXLB	LEXLBLS
G$PDDERR	TOOLPL	$PDDERR	PDDERLS
G$PERFLB	UTILPL	$PERFLB	PERFLLS
G$RATDEF	TOOLPL	$RATDEF	RATDFLS
G$RATLIB	TOOLPL	$RATLIB	RATLBLS
G$RATXVS	TOOLPL	$RATXVS	RATXVLS
G$SID	SIDPL	$SID	SIDLS
G$SYSDEF [2,3,5]	SYSDFPL,COSPL	$SYSDEF	SYDEFLS
G$SYSTXT [2]	SYSDFPL,COSPL	$SYSTXT	SYTXTLS
G$TDGTOC	TOOLPL	$TDGTOC	TDGTOLS
G$UTLDEF [2,3,5]	COSPL,SYSDFPL	$UTLDEF	UTDEFLS
G$UTLTXT [2]	COSPL,SYSDFPL	$UTLTXT	UTTXTLS
G$WTOOLS	TOOLPL	$WTOOLS	TOOLSLS
G$YYPLB	TOOLPL	$YYPLB	YYPLBLS
GADSTAPE	LDRPL	ADSTAPE	ADSTPLS
GAPML	CALPL	APML	APMLLS
GAUDPL	UPDPL	AUDPL	AUDPLLS
GBIND	LDRPL	BIND	BINDLS
GBLOCK	TOOLPL	BLOCK	BLOCKLS
GBUILD [5]	LDRPL	BUILD	BUILDLS
GCAL [5]	CALPL	CAL	CALLS
GCOMPARE	UTILPL	COMPARE	COMPLS
GCOPYD [1]	UTILPL	COPYD	COPYDLS
GCOPYF [1]	UTILPL	COPYF	COPYFLS
GCOPYNF	UTILPL	COPYNF	COPYNLS
GCOPYR [1]	UTILPL	COPYR	COPYRLS
GCOPYU	UTILPL	COPYU	COPYULS
GCOSTXT [2]	COSPL,SYSDFPL	COSTXT	CSTXTLS
GDDA [4]	DBGPL	DDA	DDALS
GDEBUG [4]	DBGPL	DEBUG	DEBUGLS
GDRD [4]	DBGPL	DRD	DRDLS
GDSDUMP	UTILPL	DSDUMP	DSDMPLS
GFLODUMP	UTILPL	FLODUMP	FLOWDLS
GFTREF	TOOLPL	FTREF	FTREFLS
GITEMIZE	UTILPL	ITEMIZE	ITEMLS

1 Procedure has an ALLOC parameter
2 Procedure has a CL parameter
3 Procedure has a default DEF=CALV2
4 Procedure has a default DEF=COS:XMP
5 Procedure has a TYPE parameter

Procedure	PL Name	Binary	Listing
GLD2	LDRPL	LD2	LD2LS
GLDR	LDRPL	LDR	LDRLS
GMODECKS	UPDPL	MODECKS	MDCKSLS
GMODSEQ	TOOLPL	MODSEQ	MODSQLS
GMODSET	TOOLPL	MODSET	MODSTLS
GMTDUMP	UTILPL	MTDUMP	MTDMPLS
GNOTE	TOOLPL	NOTE	NOTELS
GNUPDATE	NUPDPL	NUPDATE	NUPDLS
GOLDCAL	CALPL	OLDCAL	OCALLS
GPERFMON	TOOLPL	PERFMON	PFMONLS
GPREMULT	PMULTPL	PREMULT	PMULTLS
GQSTSUI	TOOLPL	QSTSUI	QSTSULS
GQUERY	UTILPL	QUERY	QUERYLS
GSEGLDR [5]	LDRPL	SEGLDR	SEGLDLS
GSEGRLS	LDRPL	SEGRLS	SEGRLLS
GSKIPD [1]	UTILPL	SKIPD	SKIPDLS
GSKIPF [1]	UTILPL	SKIPF	SKIPFLS
GSKIPR [1]	UTILPL	SKIPR	SKIPRLS
GSKIPU	UTILPL	SKIPU	SKIPULS
GSKOL	SKOLPL	SKOL	SKOLLS
GSKOLREF	SKOLPL	SKOLREF	SKREFLS
GSKOLTXT	SKOLPL	SKOLTXT	SKTXTLS
GSORT	UTILPL	SORT	SORTLS
GSPAWN	TOOLPL	SPAWN	SPAWNLS
GSPY	UTILPL	SPY	SPYLS
GSTEP	TOOLPL	STEPL	STEPLS
GSYSREF	TOOLPL	SYSREF	SYREFLS
GTARGET	UTILPL	TARGET	TARGELS
GTEDI	TEDIPL	TEDI	TEDILS
GUNB	UTILPL	UNB	UNBLS
GUNBLOCK	TOOLPL	UNBLOCK	UNBLKLS
GUPDATE [5]	UPDPL	UPDATE	UPDATLS
GUPIC	UPDPL	UPIC	UPICLS
GWRITEDS	UTILPL	WRITEDS	WRTDSLS

The QSTSTRT procedure is also defined in PRDPROCS. This procedure is used by the GENTOOL job to generate a minimal set of software tools for sites that do not have any. These tools are then used to generate the full set of software tools.

---

1 Procedure has an ALLOC parameter
5 Procedure has a TYPE parameter

1.3.15 SLPROCS GENERATION PROCEDURES

The following are the SUPERLINK generation procedures defined in the common deck SLPROCS in GENPL:

Procedure	PL Name	Binary	Listing
G$SLLIB [1]	SLLIBPL	$SLLIB [2,3]	SLLIBLS
GAACDEF	SLLIBPL	AACDEF [3]	(none)
GFRLS	SLLIBPL	FRLS [3]	FRLSLS
GFSELECT	SLLIBPL	FSELECT [3]	FSELLS
GRELEASE	SLLIBPL	RELEASE [3]	RLSLS
GREWIND	SLLIBPL	REWIND [3]	REWLS
GSLSUB	SLLIBPL	SLSUB [2]	SLSUBLS

1.4 GENPL MISCELLANEOUS PROCEDURES

Several miscellaneous and low-level procedures are defined throughout GENPL. These procedures are used by either the G*name* (high-level) generation procedures, the generation jobs, or as operational aids. There is no standard set of parameters associated with these procedures. They vary according to the function they perform. Most procedures contain parameters for ID and owner.

The defaults for these parameters are taken from the ID and OWN common decks, but site-specific values may be substituted to customize the generation process.

There are several procedures defined in the common deck MSCPROCS that are not used during system generation, but are helpful aids. These procedures are described in the following paragraphs.

The CF77 procedure invokes the Fortran Autotasking system, FPP, FMP, CFT77, and SEGLDR. See the COS Autotasking User's Guide, publication SN-3033, for more information on CF77.

DSDISK generates a deadstart IOS expander disk directory. It assumes that the datasets it uses are local to the job. The GENDSDK job provided in GENPL accesses these datasets and calls the DSDISK procedure to create a directory with the name in the IOSDIR common deck in GENPL. The IOSDIR common deck in the released PL contains a call to the ID common deck. The following is a list of the parameters, along with the default values and descriptions, used by the DSDISK procedure:

---

1 Procedure has an ALLOC parameter
2 Binary dataset name is prefixed with 'SLMULTI'
3 Binary dataset name is prefixed with 'SLSTACK'

Parameter	Default	Description
CLEARIO	CLEARIO	Name of CLEARIO dataset in the expander disk directory
COS	COS	Name of COS dataset in the expander disk directory
DISKLD	DISKLD	Name of $DISKLD dataset in the expander disk directory
DSDIAG	DSDIAG	Name of DSDIAG dataset in the expander disk directory
DUMP	DUMP	Name of $DUMP dataset in the expander disk directory
KERNEL	KERNEL	Name of $KERNEL dataset in the expander disk directory
MF	AP	Expander mainframe identifier
PARAMS	RESTART	Name of STARTUP parameter file in the expander disk directory
TAPELD	TAPELD	Name of $TAPELD dataset in the expander disk directory
TEXT	DSDISK	Name of expander disk directory

The DSTAPE procedure generates an IOS deadstart expander tape. It assumes that the datasets it uses are local to the job. The GENDSTP job provided in GENPL accesses these datasets and calls the DSTAPE procedure to create a deadstart tape with the name in the IOSTAPE common deck in GENPL. The IOSTAPE common deck in the released PL contains a call to the ID common deck. The following is a list of the parameters, along with the default values and descriptions, used by the DSTAPE procedure:

Parameter	Default	Description
MF	AP	Expander mainframe identifier
TEXT	DSTAPE	Name of expander tape

Both the GENDSTP and GENDSDK jobs assume that the site has created a permanent dataset named PARAMS with an ID and OWN matching these common decks in GENPL.

The GETTXT procedure accesses the OLDCAL system text files necessary to establish a multitasking, stack calling sequence, SUPERLINK multitasking, or SUPERLINK stack calling sequence environment for a job. The following is a list of parameters, along with the default values and descriptions, used by the GETTXT procedure:

Parameter	Default	Description
ED	ED common deck	Permanent dataset edition
ID	ID common deck	Permanent dataset identifier
MULTI†	None	Multitasking environment
NA†	None	No abort or ACCESS
SLMULTI†	None	SUPERLINK multitasking environment
SLSTACK†	None	SUPERLINK Stack environment
STACK†	None	Stack environment
OWN	OWN common deck	Permanent dataset owner

The MULTI, SLMULTI, SLSTACK, and STACK parameters are mutually exclusive.

The MULTI procedure accesses the datasets necessary to establish a multitasking environment for a job. The STACK procedure accesses the datasets necessary to establish a stack calling sequence environment for a job. The following is a list of the parameters, along with the default values and descriptions, used by these procedures:

Parameter	Default	Description
ED	ED common deck	Permanent dataset edition
ID	ID common deck	Permanent dataset identifier
NA†	None	No abort or ACCESS
OWN	OWN common deck	Permanent dataset owner

SLMULTI and SLSTACK are procedures defined in the common deck SLPROCS in GENPL that are not used during system generation, but are helpful aids. The SLMULTI procedure accesses the datasets necessary to establish a SUPERLINK multitasking environment for a job. The SLSTACK procedure accesses the datasets necessary to establish a SUPERLINK stack calling sequence environment for a job. The following is a list of the parameters, along with the default values and descriptions, used by these procedures:

Parameter	Default	Description
ED	ED common deck	Permanent dataset edition
ID	ID common deck	Permanent dataset identifier
NA†	None	No abort or ACCESS
OWN	OWN common deck	Permanent dataset owner

† Represents an unequivalenced parameter

Several procedures defined in the common deck MSCPROCS are used by the generation jobs to access datasets and establish a build environment during system generation. These procedures are described below.

The GETBIN procedure tries to find a permanent dataset in the following order:

1. As a permanent dataset under the ID, ED, and OWN parameters.

2. As a temporary dataset under the ID, ED, and OWN parameters. If the BASE parameter is specified, the following steps are performed:

   - As a permanent dataset under the BASEED, BASEID, and BASEOWN parameters.

   - As a temporary dataset under the BASEED, BASEID, and BASEOWN parameters.

The following is a list of the parameters, along with the default values and descriptions, used by the GETBIN procedure:

Parameter	Default	Description
BASE†	(none)	Use the BASE parameters
BASEED	BASEEP common deck	Permanent dataset edition to use as a base
BASEID	BASEID common deck	Permanent dataset identifier to use as a base
BASEOWN	BASEOWN common deck	Permanent dataset owner to use as a base
DN	(required)	Local dataset name
ED	ED common deck	Permanent dataset edition
ID	ID common deck	Permanent dataset identifier
NA†	(none)	No abort on ACCESS
OWN	OWN common deck	Permanent dataset owner
PDN	(none)	Permanent dataset name

† Represents an unequivalenced parameter

The GETBAS procedure accesses datasets to form a base for building. The LEVEL parameter determines how extensive of a base is accessed. If LEVEL is greater than or equal to 1, the products built in the basic utility generation job series are accessed. If LEVEL is greater than or equal to 2, ADSTAPE, APML, BIND, LDR, NUPDATE, SYSREF and UNB are also accessed. The GETBAS procedure invokes the GETBIN procedure to access all of its datasets, so the BASE, BASEED, BASEID, and BASEOWN parameters are used by it also. The following is a list of the parameters, along with the default values and descriptions, used by the GETBAS procedure:

Parameter	Default	Description
BASE†	(none)	Use the BASE parameters
BASEED	BASEED common deck	Permanent dataset edition to use as a base
BASEID	BASEID common deck	Permanent dataset identifier to use as a base
BASEOWN	BASEOWN common deck	Permanent dataset owner to use as a base
ED	ED common deck	Permanent dataset edition
ID	ID common deck	Permanent dataset identifier
LEVEL	1	Basic level indicator
NA†	(none)	No abort on ACCESS
OWN	OWN common deck	Permanent dataset owner

The GETPL and GETNPL procedures access a program library with a permanent dataset name equal to the NAME parameter followed by a PL. GETPL is used for UPDATE format PL. GETNPL is used for a NUPDATE format PL. The procedure then tries to locate a local modification (LM) dataset (NAME parameter followed by a LM) either as a local or permanent dataset. If it finds a LM dataset, it updates the PL using the LM dataset as an input dataset. If the DN parameter is an asterisk (*), the updated PL local

† Represents an unequivalenced parameter

dataset name is the same as its PDN. The following is a list of the parameters, along with the default values and descriptions, used by these procedures:

Parameter	Default	Description
BASE†	(none)	Use the BASE parameters
BASEED	BASEED common deck	Permanent dataset edition to use as a base
BASEID	BASEID common deck	Permanent dataset identifier to use as a base
BASEOWN	BASEOWN common deck	Permanent dataset owner to use as a base
DEF	(none)	Symbols to be defined for UPDATE
DN	*	Local PL dataset name
ED	ED common deck	Permanent dataset edition
ID	ID common deck	PL permanent dataset identifier
LMABORT†	NA	Abort on LM dataset ACCESS
LMID	ID common deck	LM permanent dataset identifier
LMOWN	OWN common deck	LM permanent dataset owner
NA †	(none)	No abort on PL dataset ACCESS
NAME	(required)	PL permanent dataset name (PL name minus PL)
OWN	OWN common deck	PL permanent dataset owner

The START procedure is invoked at the end of each job in the basic utility generation job series if the SW-SPAWN flag is enabled. START invokes GETPL with NAME=GEN, then performs the appropriate UPDATE and SUBMIT to start executing a generation job. This procedure may be used to start executing any job in GENPL. The following is a list of parameters, along with the default values and descriptions, used by the START procedure:

† Represents an unequivalenced parameter

Parameter	Default	Description
DEF	(none)	Symbols to be defined for UPDATE
DID	(none)	Destination mainframe identifier
JOB	(required)	GENPL job to start executing
NRLS†	(none)	GENPL job is not released after the SUBMIT
SID	(none)	Source front-end system identifier

The ERASE procedure is used by the DELTEMP generation job to delete temporary versions of datasets after the final versions have been generated. The following is a list of parameters, along with the default values and descriptions, used by the ERASE procedure:

Parameter	Default	Description
DN	(required)	Name of dataset to delete
ED	ED common deck	Permanent dataset edition
ID	ID common deck	Permanent dataset identifier
OWN	OWN common deck	Permanent dataset owner

The ERASEPL procedure is used by some of the generation jobs to delete the Cray proprietary PLs after the software has been generated. These PLs cannot be left on the system. The following is a list of parameters, along with the default values and descriptions, used by the ERASEPL procedure:

Parameter	Default	Description
ED	ED common deck	Permanent dataset edition
ID	ID common deck	Permanent dataset identifier
NAME	(required)	Name of PL to delete
OWN	OWN common deck	Permanent dataset owner

Several procedures are available to save different types of datasets with different permanent dataset names or IDs. The names of these procedures all begin with SAVE followed by an abbreviation of one of the following dataset types: BIN (binary), CBIN (C 4.0 binary), CLST (C 4.0 listing), LIST (listing), MLST (multitasking listing), MULT (multitasking), SLST (stack listing), STCK (stack), TEMP (temporary). The following is a list of the parameters, along with the default values and descriptions, used by these procedures:

† Represents an unequivalenced parameter

Parameter	Default	Description
DN	(required)	Local dataset name
ED	ED common deck	Permanent dataset edition
ID	ID common deck	Permanent dataset identifier
PDN	(none)	Permanent dataset name
REPLACE†	REPLACE common deck	Existing system binary is deleted before the new one is saved
SL	(none)	Permanent dataset name prefix

The PDN parameter is available only for BIN, CBIN, CLST, and LIST dataset types. The SL parameter is available only for MLST, MULT, SLST and STCK dataset types.

1.5 DGS SOFTWARE GENERATION

The following subsections give the steps for the generation of the ECLIPSE local station, provided by CRI. The ECLIPSE station can be configured with either a TEC or an AMPEX display terminal. The release tape provides procedures for generating the station.

Enter commands, except where noted, at the Data General master console. These commands may require appropriate responses to prompts displayed by the procedure file (see the Data General Station (DGS) Operator's Guide).

Procedures described in this subsection generate ECLIPSE stations, which log memory errors produced by Cray computer systems reporting either 36 or 32 bits of error code. Table 1-1 gives the generation procedure and station binary file names of valid combinations of station types and characteristics.

(Systems reporting 36 bits must have the CRAY-1 36-bit error channel to the maintenance control unit (MCU) and hardware modules ZJ and ZQ. These systems are defined as the following: CRAY-1 serial numbers 18, 26, and up; any CRAY-1 S computer system upgraded to a model 1200 system or above; and any CRAY X-MP or CRAY-1 M model.)

† Represents an unequivalenced parameter

Table 1-1. Generated Stations, Procedures, and Characteristics

Local Station Characteristics			
TEC Display		AMPEX Display	
32-bit Error Code Logging	36-bit Error Code Logging	32-bit Error Code Logging	36-bit Error Code Logging
GENCON32.MC ESTAT.<OL,SV> ESTATF.<OL,SV>	GENCON36.MC STAT.<OL,SV> STATF.<OL,SV>	GENALOC32.MC EXSTAT.<OL,SV> EXSTATF.<OL,SV>	GENALOC36.MC XSTAT.<OL,SV> XSTATF.<OL,SV>

After creating the generation directory and loading the proper files from the release tape, the building of any station involves the following steps:

1. Execute the appropriate procedure to create the station binaries.

2. Bring up the station under the generation directory to verify that both background and foreground versions run.

3. Move the station into the master directory.

Local station generation procedures change often. As an aid to the analyst, a copy of this subsection is available on the station release tape as file GENDOC (GENDOC is part of tape file 12, Mods and Generation Procedures).

To generate a local station, perform the following steps (press the carriage return after entering the command):

1. To create a directory for the local station generation, enter the following commands:

```
CDIR CRI
DIR CRI
```

2. Load the tape that contains the generation procedures onto the ECLIPSE:

   a. Mount the local station release tape on unit 0
   b. Enter the following commands:

```
INIT MT0
LOAD/V MT0:(12,13)
RELEASE MT0
```

Generate the 32- or 36-bit local station using the procedures in the following subsections.

1.5.1 32-BIT STATION GENERATION

The 32-bit stations are used on Cray computer systems with 1 million words of memory or less.

After completing the steps to create a directory for local station generation and loading procedure and source files, perform the following steps:

1. Verify that all necessary station generation files are present. This generation uses the following files:

Files	Description
DBLKLOC.MC	Deblocks station binary files
GENLOC32.MC	Assembles the local 32-bit stations
LINKS.MC	Links RDOS files to the subdirectory being used for station generation.
L$SOURCE.FL	Current list of station deck names
L$ESTAT.CM	Generates the TEC 32-bit background station binaries
L$ESTATF.CM	Generates the TEC 32-bit foreground station binaries
L$EXSTAT.CM	Generates the AMPEX 32-bit background station binaries
L$EXSTATF.CM	Generates the AMPEX 32-bit foreground station binaries

2. Bring up an RDOS station and save the station PL, RDLOCPL (ID=CRI), on the Cray mainframe.

3. Modify LOCJOB to have the proper account information for your site and submit the job to run on the Cray mainframe; LOCJOB stages about 91 files back to RDOS. If permanent dataset privacy is turned on, modify the ACCESS information of RDLOCPL to include the ownership value.

4. Verify that all necessary station files have staged from the Cray mainframe back to RDOS. The current files are:

E$GLOBA.BL	L$AT.BL	L$BABEL.BL	L$BYTES.BL
L$C00.BL	L$C01.BL	L$C02.BL	L$C03.BL
L$C04.BL	L$C05.BL	L$C06.BL	L$C07.BL
L$C08.BL	L$C09.BL	L$C10.BL	L$C10X.BL
L$C11.BL	L$C11X.BL	L$C12.BL	L$C13.BL
L$C14.BL	L$C15.BL	L$C16.BL	L$C17.BL
L$C18.BL	L$C19.BL	L$C20.BL	L$C21.BL
L$C22.BL	L$C23.BL	L$C24.BL	L$C25.BL
L$C26.BL	L$C27.BL	L$C28.BL	L$C29.BL
L$C30.BL	L$CLASS.BL	L$CLASX.BL	L$COMMA.BL
L$CONSO.BL	L$CRAY.BL	L$DATA.BL	L$DEBUG.BL
L$DECOD.BL	L$DISPL.BL	L$DUMP.BL	L$ER1.BL
L$ER2.BL	L$ER3.BL	L$ERROR.BL	L$FATAL
L$FILE.BL	L$FORMA.BL	L$GLOBA.BL	L$INIT.BL
L$INPUT.BL	L$INTER.BL	L$KEYWO.BL	L$LINK.BL
L$LOGOF.BL	L$LOGON.BL	L$MATCH.BL	L$MESSA.BL
L$MODIF.BL	L$MODIX.BL	L$OUTPU.BL	L$OVERL,BK
L$PARIT,BK	L$PROTO.BL	L$QUEUE.BL	L$READ.BL
L$REFRE.BL	L$REQUE.BL	L$ROLL.BL	L$SCAN.BL
L$SELEC.BL	L$SPACE.BL	L$START.BL	L$STATE.BL
L$STATI.BL	L$STATU.BL	L$STATX.BL	L$STMSG.BL
L$STORA.BL	L$SYNTA.BL	L$SYSDU.BL	L$TIMER.BL
L$TITLE.BL	L$UTILI.BL	X$GLOBA.BL	

5. Enter the following commands on the master console to generate the 32-bit local TEC and AMPEX stations. These procedures will ask the operator to press any key to start the generation procedure:

```
DBLKLOC
GENLOC32
```

The 32-bit local stations are now generated.

6. To bring up a 32-bit error logging station, enter at the Data General master console the appropriate background and foreground commands from the following list:

Command	Description
ESTAT	TEC background local station
EXSTAT	AMPEX background local station
EXFG ESTATF	TEC foreground local station
EXFG EXSTATF	AMPEX foreground local station

1.5.2 36-BIT STATION GENERATION

Use the 36-bit station with Cray mainframes that have more than 1 million words of memory.

After you complete the steps to create a directory for local station generation and load procedure and source files, perform the following steps:

1. Verify that all necessary station generation files are present. This generation uses the following files:

Files	Description
DBLKLOC.MC	Deblocks station binary files
GENLOC36.MC	Assembles the local 36-bit stations
LINKS.MC	Links RDOS files to the subdirectory being used for station generation
L$SOURCE.FL	Lists current station deck names
L$STAT.CM	Generates the TEC 36-bit background station binaries
L$STATF.CM	Generates the TEC 36-bit foreground station binaries
L$XSTAT.CM	Generates the AMPEX 36-bit background station binaries
L$XSTATF.CM	Generates the AMPEX 36-bit foreground station binaries

2. Bring up an RDOS station and save the station PL, RDLOCPL, on the Cray mainframe.

3. Modify LOCJOB to have the proper account information for your site and submit the job to run on the Cray mainframe; LOCJOB stages about 91 files back to RDOS.

4. While you wait for LOCJOB to finish staging the files to RDOS, print copies of the files named in step 1.

5. Verify that all necessary station files have staged back to RDOS. Each staged file's name ends with the suffix .BL and should correspond to the list of names in L$SOURCE.FL (L$SOURCE.FL lists the files without the .BL suffix, however).

6. Enter the following commands on the master console to generate the 36-bit local TEC and AMPEX stations. These procedures ask the operator to press any key to start the generation procedure.

```
DBLKLOC
GENLOC36
```

The 36-bit local stations are now generated.

7. To bring up a 36-bit error logging station, enter at the Data General master console the appropriate background and foreground commands from the following list:

Command	Description
STAT	TEC background local station
XSTAT	AMPEX background local station
EXFG STATF	TEC foreground local station
EXFG XSTATF	AMPEX foreground local station

2. GENERATING SITE-SPECIFIC COS SOFTWARE

The released COS binaries are designed to run on a minimum hardware configuration. Each site has its own hardware configuration and preferences for how the system should schedule jobs, allocate resources, and charge for resources used. The modifications that change these variables are called local configuration modifications (mods). Mods usually are changes to table entries and to values of symbols called installation parameters that control the way the system runs. Local configuration modifications are not the same as mods that add a new feature or change the way a feature is implemented. Three main methods are available for changing the system:

- The first method makes changes to table entries and installation parameters, generally by changing the source code with UPDATE format modifications. This section describes installation parameters and frequently changed tables.

  If your site requires changes to table entries and installation parameters, use the source listings containing the tables to determine the UPDATE directives to use to replace the released values. A set of mods should then generate the affected programs as previously described.

- The second method uses the directives in the startup parameter file at startup time. Section 5, COS Operations, describes these directives.

- The third method of changing the system is through COS debugging commands. Section 6, Station Debug Commands, describes these directives.

Choosing one of the three methods is a matter of convenience or preference; each has advantages and disadvantages in a given situation. If a table or installation parameter is rarely changed (that is, if several months pass between changes), then the UPDATE format is recommended. These changes must be planned and tested before being used in the production system. If a table or installation parameter is changed at an unpredictable time (for instance, for adding new flaws or for specifying a disk temporarily out of service), the startup directive is recommended. If tables or installation parameters need to be changed for tuning the system or isolating a problem, the debug commands are recommended. Debug commands are the least permanent and carry the greatest risk, since no record of the change appears and the effects of the commands disappear at the next startup.

2.1 CONFIGURATION PROCEDURES

The following subsections contain information and instructions designed for using and creating configuration modifications that correspond to the hardware installed at your site. To generate site-specific COS software, you will need the following:

- A description of the hardware at your site and its interconnections. Your field engineers (FEs) can provide you with this information.

- Listings with UPDATE sequence numbers of the following:

Deck Types	Deck Names	Libraries
Common	CONFIG@P	COSPL
Common	COSI@P	COSPL
Common	USERI@P	COSPL
Regular	STPTAB	COSPL
Regular	IOSDEF ($IOSDEF)	IOPPL
Regular	AMAP	IOPPL

- Additionally, but not necessarily, listings of the following:

  $SYSDEF
  $UTLDEF
  COSDEF

The following subsections provide the instructions to help configure a system:

2.2 Setting General Hardware, Memory, CPU, and Other Parameters
2.3 Disk Configuration
2.4 On-line Tape Configuration
2.5 SUPERLINK/MVS Configuration
2.6 Front-end Computer Connections Configurations
2.7 IOS Configuration
2.8 Generic Resource Configuration
2.9 Other Configuration Tables
2.10 Site-specific Target Machine Name Configuration
2.11 Table Macros and Build Table Macros
2.12 Configuration Examples

Each subsection describes which tables or macros must be redefined to properly configure your system. Subsection 2.11, Table Macros and Build Table Macros, explains in detail how to use the macros and tables to configure a system.

A number of symbols defined in COSDEF, EXEC, or STP and assembled into COS or other parts of the software can be changed according to individual site needs or preferences. These symbols are in two groups: the hardware configuration parameters and the COS parameters.

The hardware configuration parameters begin with the characters C@ and appear in deck CONFIG@P. COS parameters begin with the characters I@ and appear in decks COSI@P, USERI@P, and STPTAB.

Appendix A gives detailed descriptions of the installation parameters in CONFIG@P, COSI@P, and USERI@P.

2.2 SETTING GENERAL HARDWARE, MEMORY, CPU, AND OTHER PARAMETERS

Configuration of the hardware characteristics of the Cray computer system consists of setting the hardware parameters defined in the common deck CONFIG@P. See appendix A for a complete description of hardware parameters. For multiple CPU mainframes, an entry must be made in the Configuration Table (CNT) to allow CPU UP/DOWN configuration changes during normal operation. See subsection 2.4.1, Configuration Table (CNT), for more information on changing the CNT.

2.3 DISK CONFIGURATION

See the disk group of installation parameters found in appendix A for parameters that must be set to configure disks.

The default system has only one disk unit (the master disk) configured. For a site with an I/O Subsystem (IOS), the default master disk is on IOP-1, channel 20. For a non-IOS site, the default master disk is on Cray channel pair 2, unit 0.

The hardware configuration of disk units is communicated to COS through the Equipment Table (EQT). The information contained in the EQT is used by Startup to build the Disk Reservation Table (DRT). See the COS Table Descriptions Internal Reference Manual, publication SM-0045, for more details on these tables.

There is one EQT entry and one DRT entry for each disk unit configured. The following subsections describe these entries.

2.3.1 CONFIGURING THE EQUIPMENT TABLE (EQT)

The EQT is located in STPTAB at the label B@EQT. The EQT entries are the primary repositories of information about the disk drives. One EQT entry must be defined for each device (disk, SSD, and Buffer Memory Resident (BMR) disk) connected to the system, either physically or logically. (A striped disk group is a logical device and must have a separate EQT entry. Each disk that is part of a striped disk group is a physical device and must also have its own EQT entry.)

If more than one logical device has been defined on the same physical device using the EQT macro parameters STK and NTK, each of those devices must also have its own EQT entry.

For DD-39 Disk Storage Units, each drive is configured separately, that is, each drive has its own EQT entry that specifies the drive's unit number. Valid unit numbers for a DD-39 DSU are 0, 1, and 2.

Use the EQT macro to create each entry in the EQT. To create the EQT for your system, replace the default EQT entry with EQT entries that describe your particular configuration. See subsection 2.11.6, Equipment Table (EQT) Macro, for a full description of the available parameters.

2.3.2 CONFIGURING THE DISK RESERVATION TABLE (DRT)

The DRT is located in STPTAB at the label B@DRT. The DRT controls currently allocated and available disk space. A bit in the DRT represents one allocation unit. The DRSPB (Sectors Per Bit) field in the DRT describes the size of one allocation unit.

Space is reserved automatically by the EQT macros at assembly time for the entire DRT based on the type of each device configured and the size of that device. DRT entry space is then allocated at STARTUP time and dynamically linked with the EQT entries. This means flaws cannot be assembled into the DRT as in the past.

It is also possible to reconfigure devices and change device types at STARTUP through the parameter file or operator configuration changes.

NOTE

If changing device types increases the size of the DRT entry for that device, there must be sufficient reserved space to accommodate the increase. For example, a DD-19 is replaced by a DD-29. There may not be enough space in the DRT to contain the additional information unless another device is made unavailable, or the system has been assembled with extra space reserved for the DRT.

2.3.3 BUFFER MEMORY REQUIREMENTS FOR DISKS

The Buffer Memory requirement for disks is satisfied by the amount of Buffer Memory allocated for use to the Buffer I/O Processor (BIOP) and, if present, the Disk I/O Processor (DIOP). (See subsection 2.7, IOS Configuration, for the proper procedure.)

Each disk channel requires a number of 512-word buffers to serve as a read-ahead area in Buffer Memory. The number of buffers required depends on the device type as defined in the IOS CHANNEL macro (see subsection 2.11.4, CHANNEL macro). The following list gives the number of buffers for each device type:

Device Type	Symbols Used in Macro	Buffers per Device
DD-19	D1	3
DD-19 striped	D1S	18
DD-29	D2	3
DD-29 striped	D2S	18
DD-39	D3	25
DD-39 striped	D3S	25
DD-40	D40	49
DD-40 striped	D40S	49
DD-49	D4	43
DD-49 striped	D4S	43

To determine the amount of Buffer Memory needed for disks on an IOP, multiply the number of configured devices by the buffer requirement for that device type.

The number of sectors (512-word buffers) of Buffer Memory required for four DD-29 Disk Storage Units (DSUs) and two DD-49 DSUs (not striped) configured on BIOP is calculated in the following example:

NS (number of sectors) = Number of devices * buffers per device
NS for DD-29s = 4 * 3 = 12
NS for DD-49s = 2 * 43 = 86
NS total = NS for DD-29s + NS for DD-49s = 12 + 86 = 98

2.3.4 STRIPED DISK GROUP CONFIGURATION

Use the LDEV macro described in subsection 2.11.8 to configure striped disk groups in the IOS. These macros make entries in the AMAP overlay. Changing the TY parameter on the CHANNEL macro from D1, D2, D3, D40 to D1S, D2S, D3S, D40s or D4S, respectively, indicates that an individual unit is a member of a striped group. This sets up an appropriate read-ahead area for units in a striped group. The size of the read-ahead area allows for the rotational phase differences of the units in a group. The following is a set of rules for configuring a striped disk group:

1. All units in a striped disk group must be the same device type (that is, they must be all DD-19 DSUs, or all DD-29 DSUs, and so on). They must also all have the same logical starting track and the same logical capacity (see STK and NTK parameters in the EQT macro). All members of a DD-39 or DD-40 striped group must have the same unit number (the UNT parameter in the EQT macro).

NOTE

Striped groups of DD-19 or DD-29 DSUs must use the full physical capacity of the device. This means the default values for the STK and NTK parameters on the EQT macro must be used when striping these DSUs.

2. A striped disk group must be configured in COS as residing on IOP-0 on a channel between 20_8 and 37_8. The channel number must agree with the channel number specified on the LDEV macro (see subsection 2.11.8, LDEV Macro).

3. The number of units in a striped disk group is limited by the width of the field DA@SEC in the request. This field specifies the sector number and is currently 7 bits long so that the maximum sector number is 127. The number of units is limited to the number of times the sectors on one physical track can divide into 128. For DD-19 and DD-29 DSUs, this number is 7 (128 divided by 18 equals 7). For DD-49 DSUs, this number is 3 (128 divided by 42 equals 3). For DD-39 DSUs, this number is 5 (128 divided by 24 equals 5). For DD-40 DSUs, this number is 2 (128 divided by 48 equals 2).

4. Units within a striped disk group need not reside on the same IOP.

2.3.5 BUFFER MEMORY RESIDENT (BMR) DEVICE CONFIGURATION

Buffer Memory space may be allocated for COS dataset storage. Datasets stored in Buffer Memory are called BMR datasets. The size of the Buffer Memory allocated for datasets is a multiple of 200_8K (1/16 million) words.

2.3.5.1 COS implementation

If part of Buffer Memory was allocated for dataset storage, COS treats this portion of Buffer Memory as if it were a disk attached to the IOS. This implementation allows the user to take advantage of BMR datasets with a minimum of job control language and program modification.

Buffer Memory is configured as a device attached to a channel on IOP-0. The IOP configuration overlay AMAP, the COS Equipment Table (EQT), and the COS Disk Reservation Table (DRT) reflect the amount of Buffer Memory available for allocation. The channel number in the EQT macro must agree with the channel number specified on the LDEV macro (see subsection 2.11.8, LDEV Macro).

To configure BMR datasets, use the following method to calculate the total number of sectors available:

1. Each 1/16 Mword has 128 sectors of Buffer Memory.

2. Multiply the number of sixteenths by 128 to get the total number of sectors for the C@NSEBM installation parameter.

The following example calculates the number of sectors in 1/2 Mword:

```
NS (number of sectors) = Number of 1/16 Mwords * D'128
1/2 Mword = 8/16 Mword
NS in 8/16 Mword = 8 * 128 sectors = 1024
```

The installation parameter C@NSEBM is then set to this value; in the previous case, 1024 sectors.

2.3.5.2 IOS implementation

At IOS startup time, Buffer Memory is partitioned into an area reserved for IOS use and an area reserved for BMR datasets. The partitioning is controlled by parameters specified in the configuration overlay AMAP. The amount of Buffer Memory assigned to BMR datasets must be consistent with that designated in the C@NSEBM installation parameter.

The BMR dataset request has the same format as a disk I/O request. The IOS recognizes a request directed to the channel specified in the appropriate LDEV entry of IOP-0 as a BMR dataset request. The IOS then translates the logical BMR address to the appropriate physical BMR address and performs Buffer Memory data transfer (see the I/O Subsystem Model B Hardware Reference Manual, publication HR-0030). See subsection 2.12 for an EQT configuration example.

2.4 ON-LINE TAPE CONFIGURATION

The default system is released without on-line tapes configured. See subsection A.1.3.9, Tape Parameters, for installation parameters necessary to configure tapes.

Tape devices are managed with two tables, the Configuration Table (CNT) and the Tape Device Table (TDT). The CNT defines paths to the various tape drives while the TDT is used to manage the actual I/O to and from the drive.

2.4.1 CONFIGURATION TABLE (CNT)

The CNT contains an entry for each magnetic tape device in the system. The information in each entry includes:

- Device name
- Access path information
- Device capability information
- Generic resource name

Entries are assembled into the CNT by using the CONFIG macro to define characteristics of each device (see subsection 2.11.5, CONFIG Macro).

2.4.2 TAPE DEVICE TABLE (TDT)

The TDT contains an entry for each tape device in the system. Entries are assembled into the STP tables. See the BUILD macro in subsection 2.11.3, Tape Device Table (TDT).

2.4.3 BUFFER MEMORY REQUIREMENTS FOR ON-LINE TAPE

The Buffer Memory requirement for on-line tapes must be satisfied by the amount of Buffer Memory allocated for use by the Auxiliary I/O Processor (XIOP). (See subsection 2.7, IOS Configuration, for the proper procedure.)

The minimum amount of Buffer Memory required is twice the largest tape block size allowed by the system (I@TMBS) plus 128 sectors. As the number of active tape devices increases, the amount of buffering done for each device may decrease, depending on the amount of Buffer Memory available to the XIOP. The Tape Queue Manager (TQM) dynamically adjusts the amount of buffering for all active devices whenever an additional device becomes active or an active device is released. The maximum amount of data buffering for each device is approximately 920 Kbytes for block sizes less than one half Mbyte. You should determine whether more than the minimum amount of Buffer Memory will be beneficial based on the average number of active devices and the average tape block size used at your particular site.

2.5 SUPERLINK MVS CONFIGURATION

This subsection describes SUPERLINK MVS, the communication link between MVS and COS.

SUPERLINK MVS is subject to conditional assembly in COS. To include SUPERLINK MVS in the system, you must define the I@SLT parameter found in deck COSI@P as nonzero. Also, I@OEGNE must be nonzero. For SUPERLINK MVS library support, I@SLLIB in deck USER@P must be nonzero.

The following subsections describe how to configure SUPERLINK/MVS.

2.5.1 SUPERLINK MVS SUBSYSTEM AND LIBRARY CONFIGURATION

Configuring SUPERLINK MVS involves setting tables and parameters. The following sections describe how to configure the tables. See the SUPERLINK MVS parameters in appendix A for information on setting the parameters.

Configuration may be broken down into network and link configuration. The following subsections describe network and link configuration.

2.5.1.1 Network configuration

Network configuration consists of setting the following tables:

- The local network address (TL@LAD)
- NSAP Activity Table (NAT)
- Network Routing Table (NRT)
- Network Node Table (NNT)

Local network address - The local network address is a short character string that tells SUPERLINK MVS its address on the network. It is slightly more than the name of the node in which SUPERLINK MVS resides, and is constructed in STPTAB using the BUILD macro (table prefix ONW) at label TL@LAD. The following is a listing of how to use the BUILD macro to construct the local network address:

Format:

Location	Result	Operand
	BUILD	ONW,LE,(LI=6,AFI=x'49,OI=0,NN='*nn*'L,NS=0)

NN=*nn* Node name

NSAP Activity Table (NAT) - The NAT contains an entry for each transport service in COS. There is only one transport service, therefore, the NAT only needs to be configured with one entry. Use the BUILD macro to configure this table. The following shows the BUILD macro format for configuring this table:

Format:

Location	Result	Operand
	BUILD	ONA,(LH@NAT)(TN='NAT'1,NE=1)
	BUILD	ONA,(LE@NAT)(ACB=0)

Network Routing Table (NRT) - The NRT contains an entry for each node in the network, including the Cray computer system. Each entry points to either the NAT, in the case of the local node, or a Network Node Table describing all available routes to that node.

All nodes in your installation and all routes available to the nodes that will run SUPERLINK MVS must be configured. (A node is a machine that is running SUPERLINK MVS software. Each node has a unique network address.)

The NRT contains two entry classes. The first class, of which there is only one instance, is for the local node (the Cray mainframe). This entry class contains the node name and a pointer to the NAT for this node. The second class is for remote nodes. There is one second class entry per remote node and it points to the NNT describing all routes to that node. Use the BUILD macro to configure this table. The following is the format for configuring this table with the BUILD macro:

Format:

Location	Result	Operand
	BUILD	ONR,(LH@NRT),(TN='NRT'L,NE=1)
	BUILD	ONR,(LE@NRT),(DNA=*dna*,REM=*rem*,TBA=*tba*)

DNA=*dna* Name of the node being described

REM=*rem* Remote node identifier:
- 0 Local node (CX, the Cray mainframe)
- 1 Remote nodes

TBA=*tba* Associated table base address. Points to the NAT for the local node (REM=0) or the NNT for remote node (REM=1)

Network Node Table (NNT) - The Network Node Table (NNT) describes a set of routes leading to a remote node or an intermediate node en route to a destination node.

Each entry in an NNT describes a single hardware device connected to the local node. This hardware device must also be configured in the Link Path Table (LPT) and will have an associated LPT entry ordinal.

There are two types of devices that may be configured:

- A Cray Front-end interface (FEI) or a Data Streaming interface (DSI) device
- An NSC HYPERchannel adapter

Each FEI/DSI is described with an LPT entry. However, the LPT entry for a HYPERchannel only specifies the local adapter. In the NNT, it is necessary to configure the LPT ordinal for the local adapter and the adapter address of the destination adapter.

Use the BUILD macro to configure the NNT table. The following is the format for configuring this table with the BUILD macro:

Format:

Location	Result	Operand
	BUILD	ONN,(LH@NNT),(TN='NNT'L,NE=1)
	BUILD	ONN,(LE@NNT),(LPO=*lpo*,ADR=*adr*)

LPO=*lpo* The ordinal of LPT entry describing the hardware connection to the Cray computer system

ADR=*adr* The 16-bit remote adapter address for the NSC HYPERchannel only

2.5.1.2 Link configuration

Each physical device connected to the IOS that is to be used by SUPERLINK MVS must be configured in the Link Path Table (LPT).

SUPERLINK MVS supports the following device types:

- FEI or DSI devices
- NSC HYPERchannel

Also, FEI/DSI may be operated either singly, using a half-duplex protocol, or in groups using a simplex protocol. The NSC HYPERchannel always operates using a full-duplex protocol. Where an FEI is operated with the half-duplex protocol, it may be used for both input and output, under the control of SUPERLINK MVS. Only one FEI is required to connect the machines.

If the simplex protocol is required, each FEI can operate in one direction only, so it is dedicated to either input or output. The LPT specifies in which direction an FEI operates.

Link Path Table (LPT) - Each physical device connected to the IOS that is to be used by SUPERLINK MVS must be configured in the Link Path Table (LPT). Use the BUILD macro to configure these devices. The following is the format for configuring this table with the BUILD macro:

Format:

Location	Result	Operand
	BUILD	OLP,(LH@LPT),(TN='LPT'L,NE=NE@LPT)
	BUILD	OLP,(LE@LPT),(MOD=*mod*,TYP=*typ*,ON=*on*
		ICH=*ich*,OCH=*och*,NSD=*nsd*,IBZ=*ibz*
		OBZ=*obz*,NAM=*name*)

NE@LPT=*n* *n* is the number of LPT entries. The number of LPT entries must be specified with the label NE@LPT.

MOD=*mod* The device operational mode. Use one of the following codes:

OLMO$ISM	Simplex, used for input to the Cray mainframe only (FEI/DSI only)
OLMO$OSM	Simplex, used for output from the Cray mainframe only (FEI/DSI only)
OLMO$HDP	Half-duplex, used alternately for input and output (FEI/DSI only)
OLMO$FDP	Full-duplex, used simultaneously for input and output (NSC HYPERchannel only).

TYP=*typ* Hardware type; must be one of the following:

OLCT$FEI	FEI box
OLCT$DSI	DSI box
OLCT$NSC	NSC HYPERchannel adapter

ON=*on* Set to 1 if the channel is initially active; set to 0 if it is inactive.

ICH=*ich* The input channel number of the IOP connection to the Cray computer system. The format for the IOP channel number field is as follows:

Bits	Description
0-6	Mainframe channel number to which IOP is attached
7-9	Must be zero

OCH=*och* The MIOP channel number to which the device is connected

NSD=*nsd* The 16-bit local adapter address. If this field is specified, it is assumed to hold a correct physical adapter address (first 8 bits) and a usable logical path identifier (last 8 bits).

If this field is not specified, the physical address is obtained from the device and the logical path identifier is assigned to SLT by the I@SLNLP installation parameter.

IBZ=*ibz* Input buffer size, in words. Used in defining simplex mode FEI devices that handle incoming data, and full-duplex mode NSC devices.

OBZ=*obz* Output buffer size, in words. Used in defining simplex mode FEI devices that handle outgoing data, and half-duplex mode FEI devices.

NAM=*name* A unique 7-character name used by the operator for configuration purposes (the DNV=*name* on the CONFIG command).

2.5.1.3 SUPERLINK MVS library support

SUPERLINK MVS library support provides record level access to data on an IBM or IBM-compatible MVS system for the CFT, CAL, Pascal, and C user.

A SUPERLINK MVS dataset is identified with the use of the FSELECT control statement, or calls to the library routines from within a program. Conditional code in $SYSLIB and $IOLIB gives control to code in the SUPERLINK library $SLLIB for I/O on a SUPERLINK MVS dataset. See the SUPERLINK MVS and COS Installation, Tuning, and Customization Guide, publication SI-0180, for configuration information of SUPERLINK MVS libraries.

2.6 FRONT-END COMPUTER CONNECTIONS CONFIGURATIONS

Most parameters associated with a front-end station are now sent by the station at LOGON time. COS must determine table sizes and configure the appropriate I/O channels so they can be processed accordingly. See the station parameters in appendix A for more information.

Cray channels used for front-end communication are in the Link Configuration Table (LCT). Each CPU channel used for front-end communication requires one LCT entry. The installation parameter I@NSTCH determines the number of LCT entries. LCT entries are defined in STPTAB after the label B@LCT.

The default system has one LCT entry, configured for an IOS Maintenance Control Unit (MCU) on channel pair 4. A site may need to replace or add to this entry.

An IOS should have an LCT entry (usually channel pair 1 on CRAY-1 S/M series computer systems and channel pair 4 on CRAY X-MP series computer systems) to allocate space for all front-end stations connected to the IOS. The channel ordinal field in this LCT entry should be set to I@NSTCHO, the number of Channel Extension Table (CXT) entries. Front-end channels attached to the IOS are described when configuring the IOS software at sites with an IOS.

CRAY-1 A, B, and S series computer systems with a Data General Eclipse Maintenance Control Unit (MCU) should have an LCT entry configured on channel pair 1.

The BUILD macro defines LCT entries. Each LCT entry is 1 word long. See the description of the BUILD macro in subsection 2.11.2, Link Configuration Table (LCT). You may change an individual LCT entry at STARTUP by using the *LCT command in the parameter file (see section 5, COS Operations).

Multiple front-end IDs can be multiplexed into one ordinal. For more details on the LCT entry, see the COS Table Descriptions Internal Reference Manual.

Channel assignments for front ends within the IOS are configured by using the CHANNEL macro. VAX BI channels are configured the same as front-end concentrator channels.

If the COS installation parameter I@FIT is nonzero, then front-end connections must be configured in the restart file. See the I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051, for a description of the CONFIGURE command.

2.7 IOS CONFIGURATION

See the I/O Subsystem (IOS) Administrator's Guide, publication SG-0307, for complete information on configurating the IOS.

2.8 GENERIC RESOURCE CONFIGURATION

Generic resource names group controlled devices with common characteristics (device type, location, speed, or density) into mutually exclusive groups. By their nature, these devices are subject to regulated access by the system. Tape drives are regulated because they cannot be shared among several users simultaneously. SSDs and Buffer Memory may be regulated so that the system can ensure guaranteed access.

In resource scheduling, generic resource names are used to keep track of the number of units of each resource that a job has reserved and used. Generic resource names are also used in associating datasets with tape devices on the ACCESS control statement and with mass storage devices on the ASSIGN control statement (COS Ready Reference Manual, publication SQ-0023, describes the control statements). The default generic resource name, *TAPE, identifies devices capable of 6250 b/i or 1600 b/i (dual density), but installations can define their own generic resource names according to specific site needs. COS supports up to 16 generic resource names. See subsection 2.11.1, Generic Resource Table (GRT).

2.9 OTHER CONFIGURATION TABLES

Modifying the following tables is not necessary to bring your system up and running, but you may find it is necessary to change the sizes of some of them if your system approaches the default limits set.

2.9.1 DATASET ALLOCATION TABLE (DAT)

The DAT contains entries for active system datasets and spooled datasets. It must be large enough to contain (temporarily) entries for datasets being saved or deleted, for all datasets in the System Directory (SDR), and for datasets in the input or output queues. If the DAT is not large enough to contain all the entries, it can be enlarged by modifying NE@DAT, defined in the common deck COMDA.

2.9.2 JOB EXECUTION TABLE (JXT)

The JXT contains entries for the desired number of active jobs. If the JXT is not large enough to contain all the entries, it can be enlarged by modifying the I@JXTSIZ installation parameter defined in common deck COSI@P. I@JXTSIZ cannot be greater than 256, the maximum number of active jobs possible under COS.

2.9.3 PERMANENT DATASET TABLE (PDS)

The PDS contains entries for active permanent datasets, which include all datasets in the SDR. If the PDS is not large enough to contain all the entries, it can be enlarged by modifying NE@PDS defined in the common deck COMPD. A change to NE@PDS takes effect on the next Startup.

2.9.4 SYSTEM DIRECTORY (SDR)

The SDR contains entries for datasets defined as system datasets by the ENTER parameter on the ACCESS statement. If the SDR is not large enough to contain all entries, it can be enlarged by modifying NE@SDR defined in STPTAB.

2.9.5 SYSTEM DATASET TABLE (SDT)

The SDT contains entries for active spooled datasets. If the SDT is not large enough to contain all the entries, it can be enlarged by modifying NE@SDT defined in the common deck COMSD. The maximum number of SDTs is limited by the size of the SDQC field, currently 16 bits. Any attempt to set the NE@SDT parameter to a larger number results in an assembly error. NE@SDT must equal NE@QDT (see the following subsection for NE@QDT).

2.9.6 QUEUED DATASET TABLE (QDT)

The QDT contains entries for active spooled datasets. If the QDT is not large enough to contain all entries, it can be enlarged by modifying NE@QDT defined in the common deck COMQD. NE@QDT must equal NE@SDT.

2.9.7 TASK EXECUTION TABLE (TXT)

The TXT contains entries for the desired number of active user tasks. The number of TXT entries should always be at least as large as the number of JXT entries. By default, the TXT and JXT have the same number of entries. If multitasking is used, the number of TXT entries can be changed by modifying the I@NTXT installation parameter (defined in common deck COSI@P). I@NTXT cannot be greater than 256, the maximum number of user tasks allowed by COS.

2.9.8 REGISTERED ID TABLE (RIT)

The STP-resident Registered ID Table (RIT) contains an entry for every registered ID in the system. An ID is entered in the RIT in one of three ways:

- Reserved ID's are assembled in the RIT with only the ID and R fields set. Only privileged jobs can use those IDs.

- If a job requests to be receptive or open a communication path, either a new entry is made in the RIT or a reserved entry is completed if the ID used is not registered.

- When a job terminates, an unreserved RIT entry is zeroed, whereas in a reserved entry, only the RIJXT field is zeroed.

The first entry in the RIT (having the offset value of LH@RIT) is reserved for system use. This entry must be present in the first position for the management of queued message buffers. The installation parameter I@MIJID determines the maximum number of RIT entries.

2.9.9 PATH TABLE (IPT)

The STP-resident IPT contains one entry for every interjob communication path established in the system. Each entry has a unique name consisting of the RIT offsets of both IDs in the path. To define the name, the smaller offset is to the left of the larger offset. The installation parameter I@MIJPA determines the maximum number of IPT entries.

2.9.10 USER DRIVER CHANNEL TABLE (UDT)

User driver channels configured in the system are driven by IOP driver overlays called by the driver shell in the IOS. The driver shell adds special-purpose driver software to the IOS so the drivers are as short as possible and can be installed without modifications to the IOS Kernel or other overlays. These drivers can be called by privileged COS jobs that need direct access to hardware channels, such as network executives for communication devices and protocols not supported by CRI.

Each physical user driver channel (input or output) has an entry in the UDT in STP. UDCOM uses the UDT to manage driver activity. A channel cannot be used as a user driver channel unless it is configured in the UDT and the IOS configuration deck AMAP.

The STP tasks that call entry points in UDCOM refer to user driver channels by ASCII names (up to 7 characters), not by physical or logical channel numbers. The UDT entry contains both the ASCII name (UDNAME) and the physical channel number (UDCH); this is the only location in which the name is mapped to the number. The channels were given names so user code does not have to be dependent on physical channel configurations.

The UDT contains the following fields:

Field	Description
UDNAME	Name by which the channel or channel pair is known to operators, user jobs, and COS tasks; up to 7 ASCII characters. The two physical channels of a pair can have the same name or different names. Channels with the same name can be treated as a unit for the purposes of reserving them.
UDON	UDON must be set if the channel is to be on logically after Startup.
UDLL	UDLL must be set if the channel is a software loop-back channel. Software loop-back channels must be built in pairs, but they need not have the same ASCII name.
UDCH	Physical channel number and location in the following format: 7/*ios*,3/*iop*,6/*num* *ios* Channel number of the low-speed channel for sending F-packets to the IOS *iop* IOP number: 0 MIOP 1 BIOP 2 DIOP 3 XIOP or second DIOP *num* Physical channel number; odd for output, even for input.
UDCCN	This is the co-channel number, or the number of the other channel in a pair of software loop-back channels. UDCCN is required only if UDLL is set. The format of UDCCN is the same as UDCH. The IOS and IOP subfields are ignored for software loop-back channels; any unique physical numbers can be given, provided they are odd numbers for output channels and even numbers for input channels, and they do not conflict with other UDT entries.

2.9.11 OTHER OPERATIONAL FEATURES

The following features can be manipulated on-site to fulfill local requirements.

2.9.11.1 System bulletin procedures

An installation can create a dataset named $BULLIT to distribute system information to users (bulletin processing). This dataset is copied to the logfile of every job.

To enable bulletin processing, the I@BULLIT installation parameter must be set to 1. The SDR is then searched for a dataset with a local dataset name of $BULLIT. If the dataset is not found, $BULLETIN is searched for as a regular permanent dataset with an ID of SYSTEM and OWN=SYSTEM. Once found, it is copied to the beginning of the user's logfile, before the installation header. If it is not found, no action is taken.

A job such as the following creates a bulletin:

```
JOB,JN=BULLETIN,P=15,US=SYSTEM.
ACCOUNT,AC=account,APW=password.
COPYF,I=$IN,O=$BULLIT.                            Read $BULLIT from $IN
SAVE,DN=$BULLIT,PDN=$BULLETIN,ID=SYSTEM.
RELEASE,DN=$BULLIT.
ACCESS,DN=$BULLIT,PDN=$BULLETIN,ENTER,ID=SYSTEM.  Enter it into SDR
/EOF
  .
  .                                               Text of bulletin
  .
/EOF
```

If $BULLIT exists and COS archiving is used, $BULLIT should be omitted from migration via the OMIT directive in the BACKUP utility. See the Operational Aids Reference Manual, publication SM-0044, for a description of the BACKUP utility.

2.9.11.2 Option procedures

The OPTION control statement allows the user to specify options, such as the number of lines per page for $OUT, for each job. A default value can be specified for each front-end station. To set a default, obtain a listing of $SYSTXT and enter the following code on the line immediately after the definition of B@FEDLP for each front end. This symbol is found in deck FEDLP in SYSDFPL.

Format:

Location	Result	Operand
	CON	'*nnn*'L+'*id*'R

nnn Number of lines per page. *nnn* must not exceed 255.

id A 2-character ID identifying the station to COS

2.10 SITE-SPECIFIC TARGET MACHINE NAME CONFIGURATION

The target machine specification facility allows users to designate that the language translators should produce code for a specific machine configuration, regardless of the machine the translator is running on. The target machine is specified by the CPU option on the language translator or TARGET control statement. The first parameter of the CPU option is the primary machine type. The basic primary machine types (CRAY-1 M, CRAY X-MP/2, CRAY X-MP/4, and so on) are predefined and are intended to be universally recognizable generic machine types.

Sites may add their own primary machine names allowing users to easily choose a particular machine configuration. The GETPMC library routine, located in $UTLIB, contains a series of tables used to define the machine type names. To define a new primary machine name, the name and machine characteristics are added to the GETPMC routine, the libraries are rebuilt, and the language translators are recompiled. New machine names should be added after UPDATE sequence number GETPMC.101.

The specific characteristics for a machine name are defined using a BUILD macro to construct a machine characteristics table. See subsection 2.12, Configuration Examples, for an example of the BUILD macro. The format follows.

Format:

Location	Result	Operand
	CON	'*sitename*'L
	BUILD	MC,SZ,(PMT='*pmt*'L,BANK=*bank*,NCPU=*ncpu*,IBSZ=*ibsz*,
		MSZ=*msz*,MSPD=*mspd*,CLK=*clk*,NCL=*ncl*,BBSY=*bbsy*,
		EMA=*option*,CIGS=*option*,VPOP=*option*,PC=*option*,
		RDVL=*option*,VRCR=*option*,AVL=*option*,HPM=*option*,
		BDM=*option*,STR=*option*,CORI=*cori*)

sitename Site-specific machine type name. This is the name the site wishes to use to define a particular machine type.

PMT=*pmt* Primary machine type; must be either CRAY-1 or CRAY X-MP computer system.

BANK=*bank* Number of memory banks

NCPU=*ncpu* Number of physical CPUs

IBSZ=*ibsz* Instruction buffer size (must be either D'16 or D'32)

MSZ=*msz*	Physical memory size - predefined symbols are available for memory sizes of: MC500K = 524,288 words MC1MEG = 1,048,576 words MC2MEG = 2,097,152 words MC4MEG = 4,194,304 words MC8MEG = 8,388,608 words MC16MEG = 16,777,216 words
MSPD=*mspd*	Memory speed (number of clock cycles per memory access)
CLK=*clk*	Clock speed (measured in picoseconds)
NCL=*ncl*	Number of cluster register sets
BBSY=*bbsy*	Memory bank busy time (in clock cycles)
EMA=TRUE FALSE	Extended Memory Addressing (EMA) hardware: TRUE EMA present FALSE No EMA
CIGS=TRUE FALSE	Compress/Index Gather/Scatter (CIGS): TRUE CIGS present FALSE No CIGS
VPOP=TRUE FALSE	Vector Population count unit: TRUE Vector Population count unit present FALSE No vector population count unit present
PC=TRUE FALSE	Programmable Clock present: TRUE Programmable clock present FALSE No programmable clock
RDVL=TRUE FALSE	Direct read vector length ability: TRUE Direct read vector length ability present FALSE No direct read vector length ability
VRCR=TRUE FALSE	Vector recursion: TRUE Vector recursion occurs (that is, if using the same vector register for input and output causes the output elements of a functional unit to be reused as input) FALSE No vector recursion

AVL=TRUE FALSE	Additional vector logical unit: TRUE Additional vector logical unit present FALSE No additional vector logical unit
HPM=TRUE FALSE	Hardware Performance Monitor: TRUE Hardware Performance Monitor present FALSE No Hardware Performance Monitor
BDM=TRUE FALSE	Bidirectional memory: TRUE Bidirectional memory present (that is, there are multiple read and write ports to memory that can operate simultaneously) FALSE No bidirectional memory
STR=TRUE FALSE	Status register: TRUE Status register present FALSE No status register
CORI=*cori*	Control operand range interrupts (ERI/DRI)

2.11 TABLE MACROS AND BUILD TABLE MACROS

COS uses the following tables for configurations. In each BUILD macro, a space must immediately follow the table name (for example, BUILD GR,(LE@GRT),(GRN='*grn*'L) where GRT is the table name).

2.11.1 GENERIC RESOURCES TABLE (GRT)

COS uses the GRT in STPTAB to define legal generic resource names.

Format:

Location	Result	Operand
	BUILD	GR,(LE@GRT),(NM='*grn*'L,WF=D'WR, AVR=*avr*)

NM=*grn* Generic resource name, 7 ASCII characters or less. This is the same name referred to in the EQT and TDT tables.

WF=D'1.5 Decimal floating-point weighing factor for SBU computation

AVR=*avr* Indicates that this device group uses automatic volume recognition. The following are the values for this parameter:
- 0 Non-AVR
- 1 AVR

2.11.2 LINK CONFIGURATION TABLE (LCT)

COS uses the LCT in STPTAB for front-end configuration.

Format:

Location	Result	Operand
	BUILD	LC,(LE@LCT),(CHN=*chn*,CHO=*cho*,CHT=*cht*,ON=*on*)

CHN=*chn* Cray CPU channel pair number
- I@IOPICH/2 For the IOS (I@IOPICH is the CPU input channel number connected to the IOS low-speed channel)

CHO=*cho* Maximum front-end ordinal associated with this channel
- 0 If the front end is connected directly to the Cray channel
- I@NSTCHO If the front ends are connected through the IOS

CHT=*cht* Channel type. Used only if CHO=0.:
- 0 IFC (channel coupler)
- 1 NSC HYPERchannel
- 2 VAX IFC (A side)
- 3 VAX IFC (B side)
- 4 GOS station (pseudo channels)

ON=*on* Indicates if the channel is initially on or off:
- 0 Off
- 1 On

2.11.3 TAPE DEVICE TABLE (TDT)

COS uses the TDT in STPTAB for tape configuration. Use the BUILD macro to define the TDT entries as follows:

Format:

Location	Result	Operand
label	BUILD	TD,(LE@TDT)

Argument:

label A label referenced by CTL=*label* in the CONFIG macro

2.11.4 CHANNEL MACRO

The IOS uses the CHANNEL macro in overlay AMAP for channel configuration.

Devices attached to the channels of a specific IOP are identified in AMAP. The CHANNEL macro defines the channel-device correspondence. The initial entry in the table is of the following form:

Format:

Location	Result	Operand
	CHANNEL	*num*

num Highest channel number described by the table

Channels from 6 to *num* are described by subsequent entries having the following form:

Format:

Location	Result	Operand
	CHANNEL	(*channel*),TY=*type*[,ORD=*ordinal*]
		[,DEST=*target*][,UN=*unit*]

channel Channel number or numbers (documentation only, the macro generates entries sequentially)

TY=*type* Type of channel or device attached:

A0	IOP-0 accumulator channel
A1	IOP-1 accumulator channel
A2	IOP-2 accumulator channel
A3	IOP-3 accumulator channel
AMPEX	AMPEX dialogue 80 display
CH	High-speed (100-Mbyte) channel
CL	CPU low-speed channel
CN	Front-end concentrator channel or VAX BI
CS	SOROC CRT
D1	DD-19 disk drive
D1S	DD-19 disk drive, member of a striped disk group
D2	DD-29 disk drive
D2S	DD-29 disk drive, member of a striped disk group
D3	DD-39 disk drive
D3S	DD-39 disk drive, member of a striped disk group
D40	DD-40 disk drive
D40S	DD-40 disk drive, member of a striped disk group
D4	DD-49 disk drive
D4S	DD-49 disk drive, member of a striped disk group
EM	Unused
ER	MIOP error logging channel
EX	Expander channel
HSX	HSX channel
NS	NSC adapter channel
UC	User channel
VA	VAX type A concentrator channel (connected to VAX physical port A)
VB	VAX type B concentrator channel (connected to VAX physical port B)
VM	VMEbus channel (FEI-3)

ORD=*ordinal*

Channel ordinal for use with TY=CN, NS, VA, VB, or VM. Channel ordinal assignments are defined in $APTEXT. The parameter values and their descriptions follow:

I@COCO	First concentrator ordinal
I@COCO+1	Second concentrator ordinal
I@CONO	First NSC concentrator ordinal
I@COVO	First VME channel ordinal
I@CPMO	First 12-byte interconnect protocol ordinal

DEST=*target*

Target memory type for use with TY=CH:

CMEM	Central Memory (default)
SSD	SSD Memory

UN=*un* DD-40 Shadow drive; highest unit number (0 or 1)

2.11.5 CONFIG MACRO

COS uses the CONFIG macro in STPTAB to configure CPUs and tape drives. Use the BUILD macro to build the CNT header with a label of B@CNT.

Format:

Location	Result	Operand
B@CNT	BUILD	CN,(LH@CNT),(SIZ=*size*,NE=*count*)

SIZ=*size* Maximum number of devices to be configured (site dependent)

NE=*count* Number of entries in use

The CONFIG macro defines the CNT entries as follows:

Format:

Location	Result	Operand
loc	CONFIG	DVN=*dvn*,GDN=*gdn*,IOP=*n*,UN=*nnn*,DT=*type*, CTL=*addr*,STS=*status*,ICH*n*=*access*, BANK=*bank*,DID=*deviceid*,CT=*ct*,LDO=*ldo*

Argument:

loc Location symbol

Parameters:

DVN=*dvn* A 1- to 8-character (ASCII) name by which the device is known to the operator

GDN=*gdn* Generic resource name; 1- to 7-character (ASCII) name by which the device can be referenced for TQM assignment. The default resource name is *TAPE, but sites may define their own resource names.

IOP=*n* IOP number to which the device is connected. Range is 0 through 3; all tape devices are connected to IOP-3 (XIOP).

UN=*nnn* Device ordinal within the TDT

DT=*type* Device type; indicates the device capabilities. The possible values for *type* are as follows:

DT@GCPE	Dual-density tape transport, with 6250 bpi (group encoded) and 1600 (phase encoded) capabilities
DT@3480	IBM 3480 magnetic tape
DT@CPU†	Central processor configuration control

CTL=*addr* Address of the corresponding TDT entry

STS=*status*

Device status. All tape devices should be configured DOWN, so the COS operator can make devices available to COS when they are not being used by other systems and to allow GRT and Statistics to be properly updated. One or more of the following can be specified, with a colon between each status (parameters in braces are mutually exclusive):

DOWN	The device is connected but unavailable
UP	The device is connected and available (tapes should not be configured up with this macro)
RDONLY	The device is available for read only
RDWRT	The device is available for reading and writing.
AVAIL	The device belongs to the Cray computer system on-line
NAVAIL	Another mainframe is using that drive (off-line).

NOTE

Configuring the devices available or not available does not affect the IOS operations. The availability of a particular drive tells TQM whether or not it can up or down a device on the IOS. There are three possible ways that drives can be configured on the IOS:

- Up/available
- Down/not available
- Down/available

---

† Only one DT@CPU entry is allowed in the CNT. All processors on a multiple processor mainframe are controlled from a single CNT entry. When DT=DT@CPU, the only other allowed parameter is DVN.

ICH*n*=*access*

n
: A number in the range 1 through 4; specifies the optional IOP channels used to access the device. For a tape device, the device is usable only if at least one channel is specified. The ICH1 keyword should be used if only one channel is connected, the ICH1 and ICH2 keywords if two channels are connected, and so on.

access
: Hardware path to the device. For a tape device, this consists of the IOP channel number (an octal number in the range 20 to 37) and a control unit identifier (a single hexadecimal digit, 0 to F). The channel and control unit can be assembled as OFF, causing the path to the device to be unavailable until turned ON through a CONFIG command from the system operator. If all paths are specified as OFF, the device entry is considered DOWN even if STS=UP is present. The formats for *access* follow, where *chan* is the IOP channel number and *cuid* is the control unit identifier:

chan
: No control unit; the channel is operative.

chan:OFF
: No control unit; the channel is inoperative.

chan:*cuid*
: Both the control unit and the channel exist and are operative.

chan:OFF:*cuid*
: Both the control unit and the channel exist but the IOP channel is inoperative.

chan:*cuid*:OFF
: Both the control unit and the channel exist but the control unit is inoperative.

chan:OFF:*cuid*:OFF
: Both the control unit and the channel exist and are inoperative.

BANK=*bank*
: Bank number where a tape device belongs (0 through 255 decimal)

DID=*deviceid*
: Physical device unit ID; a single hexadecimal digit 0 to F.

CT=*ct*
: Channel type. Values are SELECT, DTSTR, DTSTR30, and DTSTR45.

LDO=*ldo*
: Loader ordinal. Value from 0 through I@ML, corresponding to the MLT entry that is associated with this device.

2.11.5.1 Startup configuration processing

During system startup, the CNT entries are examined for validity and consistency. The following points are checked by Startup:

- The device name must be nonzero.
- The job sequence number and JXT ordinal must be 0.
- The device cannot be connected to both an IOS and a CPU channel.
- If the device is on an IOS, it must have at least one and not more than four valid IOP channel numbers.
- Control unit IDs must be valid.
- A valid device type must be specified.
- A maximum of one DT@CPU entry is present.

If Startup detects errors or inconsistencies, it posts a message to the master operator station indicating that the CNT entry for a device is invalid. The message also includes the device name given in the CNT entry. The operator must then issue a reply to Startup. Startup then requests that the operator enter configuration changes to correct the entry. See the description of the CONFIG command in the I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051, for the command format.

2.11.6 EQUIPMENT TABLE (EQT) MACRO

COS uses the EQT in STPTAB for disk configuration. There should be an EQT macro entry for each disk, BMR or SSD device defined. If more than one logical device is to be defined on the same physical device using the STK and NTK parameters, each of the logical devices must have its own EQT entry. The total number of EQT entries configured must be reflected by the installation parameter I@DD.

Format:

Location	Result	Operand
	EQT	LDV=*ldv*,DT=*dt*,CHN=*chn*,UNT=*unt*,IOP=*iop*,RBN=*rbn*,
		VOL=*vol*,SCR=*scr*,MSD=*msd*,GRN=*grn*,GRP=*grp*,NA=*na*,
		OFF=*off*,SCX=*scx*,MBT=*mbt*,STK=*stk*,NTK=*ntk*

LDV='*ldv*' The logical device name, the name shown on the STORAGE station display, and the name used by the users on the ASSIGN statement when the Request-by-name (RBN) flag is set. If this device is the dummy EQT entry for a striped disk, the name must be of the form: 'STRIPE-*n*', where *n* is the striped disk group number and the NA parameter must be present as NA=1.

DT=*dt* Device type (as defined in COMSYSEQ):

DD19	A DD-19 disk drive
DD19I	A DD-19 with CE cylinders C1 through 41_8
DD29	A DD-29 disk drive
DD29I	A DD-29 with CE cylinders C1 through 41_8
DD39	A DD-39 disk drive
DD39I	A DD-39 with CE cylinders C1 through 61_8
DD40	A DD-40 disk drive
DD40I	A DD-40 with CE cylinders C1 through 21_8
DD49	A DD-49 disk drive
DD49I	A DD-49 with CE cylinders C1 through 23_8
EBM	Buffer Memory disk
SD8	8-million-word SSD
SD16	16-million-word SSD
SD32	32-million-word SSD
SD64	64-million-word SSD
SD128	128-million-word SSD
SD256	256-million-word SSD
SD512	512-million-word SSD

CHN=*chn* Channel pair (number). The channel pair to which the device is attached. If the device is connected to the IOS, it is an IOS channel. If the device is connected to the mainframe, it is a mainframe channel. For a Buffer Memory device and a striped disk group, CHN must be the same as that specified in the LDEV Table in AMAP. This parameter is not allowed if the device type indicates an SSD.

UNT=*unt* Unit number; identifies a particular device on a shared channel (controller).

IOP=*iop* IOP number. If the disk is attached to a DCU-4 or DCU-5 controller via an IOS channel, *iop* is the number of the IOP to which the CHN (channel) is connected. For Buffer Memory and striped disks, IOP must be set to 0. This parameter is required if the disk is attached to the IOS.

RBN=*rbn* Request-by-name flag. If the RBN flag is set, this device must be specifically requested by the user on an ASSIGN statement.

0 A PUBLIC disk
1 A PRIVATE disk, must be requested by name

VOL=*vol* Volatile Device flag. Only the SSD and Buffer Memory can be flagged as volatile.

- 0 Not a volatile device
- 1 A volatile device. During STARTUP, space is reserved to dump the entire device to a nonvolatile device when the operator uses the FLUSH command.

SCR=*scr* Scratch Device flag:

- 0 Not a scratch device; datasets allocated to this device may be made permanent.
- 1 A scratch device; no permanent datasets may be created on this device.

MSD=*msd* Master Device flag:

- 0 Not the master device
- 1 This is the master device containing the Dataset Catalog (DSC) and other reserved areas. Only one master device is allowed.

GRN=*grn* Generic resource name ('*name*'L). Indicates that the device is a controlled generic resource. If GRN is present, RBN is required.

GRP=*grp* Striped-disk group number (must be 0 through 9). This disk belongs to a striped disk group. Members of a striped disk group must all have the same Device Type (DT), the same starting track (STK), and the same number of tracks (NTK) in their EQT entries. For DD-19 or DD-29 disks, the starting track must be zero, and the number of tracks must specify the full physical capacity of the DSU. Striped disk groups must be defined on the IOS by the CHANNEL and LDEV macros. A group member cannot be a master device and must be attached to an IOS channel.

NA=*na* Not available flag:

- 0 The device is available to the system
- 1 The device is unavailable to the system

OFF=*off* Unit Off flag:

- 0 Unit is on
- 1 Unit is off. No new datasets are allocated to the device. Existing datasets may be read and/or extended.

SCX=*scx* Shared channel exception flag. Indicates that the device is to be viewed by the software as residing on its own channel. Can only be specified if UNT is present and the device is on the IOS.

MBT=*mbt* Multisector transfer controller flag:

- 0 SK/SL modules are not present.
- 1 Indicates the SK/SL modules are present in a DCU-3 control unit. This allows consecutive disk sectors to be transferred without an intervening interrupt.

STK=*stk* Starting track number for a logical device in octal. Used when defining more than one logical device on a physical device. The Master Device cannot be defined on a logical device unless that device starts on track zero. Default is to start at track zero.

NTK=*ntk* The number of tracks present on the logical device in octal. Must be specified if STK is specified. The value of STK + NTK must not exceed the maximum number of tracks available on the device type defined. Default is to use the full device capacity. The following table shows the maximum number of tracks available by COS for each mass storage device:

Table 2-1. Maximum Tracks for Mass Storage Devices

Device Type	Tracks/ Device (Octal)
DD-19	10016
DD-29	20046
DD-39	10162
DD-40	64544
DD-49	15720
EBM	C@NSEBM/ O'22
SSD8	1000
SSD16	2000
SSD32	4000
SSD64	10000
SSD128	20000
SSD256	40000
SSD512	100000

2.11.7 EQTEND MACRO

COS uses the EQTEND macro in STPTAB to terminate disk configuration. An EQTEND macro call must follow the last invocation of the EQT macro.

Format:

Location	Result	Operand
	EQTEND	

There are no parameters.

2.11.8 LDEV MACRO

The IOS uses the LDEV macro in AMAP for logical device configuration.

Format:

Location	Result	Operand
	LDEV	TYPE=*typ*,CHN=*chn*,BLKS=*blk*,
		HDS=*hds*,SCS=*scs*,UNS=(*uns*)

TYPE=*typ* Type of logical device:

- BMR Buffer Memory Resident
- DSK Striped-disk group
- END End of LDEV macro; must be last.

CHN=*chn* Logical channel number between 20_8 and 37_8 inclusive. This is the value specified in the COS EQT configuration as CHN.

BLKS=*blk* Number of 200000_8K (1/16 million words) blocks of Buffer Memory assigned to this device; used with TYPE=BMR only.

HDS=*hds* Number of heads (tracks per cylinder) on each unit of a striped disk group; used with TYPE=DSK only.

SCS=*scs* Number of physical sectors per track on each unit of a striped disk group; used with TYPE=DSK only.

UNS=(*uns*)

(*ixx*,...,*ixx*) - List of the units within a striped disk group. Each entry consists of a 3-digit octal value separated by commas. The first digit specifies the number of the IOP where the disk resides; this means units within a striped disk group need not reside on the same IOP. The

UNS=(*uns*) (continued) rightmost 2 digits are the physical channel number of the disk unit. The number of units in a striped disk group is limited by the field DA@SEC in the request. This field specifies the sector number and is currently 7 bits, so the maximum sector number is 127. The maximum number of units is then defined to be 128 divided by the number-of-sectors-per-physical-track (DD-19s and DD-29s may have 7 units, DD-39s may have 5, DD-40s may have 2, and DD-49s may have 3).

Example: UNS=(122,123,233)

2.11.9 USER DRIVER CHANNEL TABLE (UDT)

The User Channel facility makes it possible for a privileged COS job to make requests of a low-speed channel driver in the IOS. It is also used by the COS task IQM to control the channels to an attached ISP.

A UDT entry is built using the following macro.

Format:

Location	Result	Operand
	BUILD	UD,(LE@UDT),(NAME=*chname*,CH=*chn*,CCN=*cochn*,ON=*on*, LL=*llflag*)

NAME=*chname*

Logical channel name. A job using a User Channel refers to it by name, not by physical channel number. This parameter also determines the DVN= parameter in the CONFIG command to turn the channel on or off. NAME must be up to 7 ASCII characters, left-justified, zero-filled. If an I/O pair are given the same logical name, they can be reserved and configured together with a single request or command.

CH=*chn* Physical channel number. Five octal digits representing the IOS, IOP number within the IOS, and channel number in the IOP. The form is SPCCC, where S represents the IOS (always one), P represents the IOP (always zero), and CCC gives the physical channel number. For example, 10030 specifies channel 30 on IOP zero (MIOP).

CCN=*cochn* Co-channel number. Five octal digits specifying a physical channel in the same format as the CN field. If two physical channels are related to each other (that is, an I/O pair to the same device), the CCN field makes this

CCN=*cochn* (continued) relationship known to the driver. Whether or not this field is required depends on the IOS driver used on the channel.

ON=*on* On/off flag:
- 0 Off
- 1 On

If a channel is built with ON=0, it may be turned on during or after startup by means of the CONFIG command.

LL=*llflag*
Local Loop-back flag (LL=1 makes the channel a local loop-back channel). Loop-back channels must be configured in pairs, one with an even CN field (input) and one with an odd CN field (output). Each must give the other's CN parameter as its own CCN. Any pair of CN values can be chosen, so long as they do not appear in any other UDT entries.

2.11.10 MEDIA LOADER TABLE (MLT)

COS uses the Media Loader Table (MLT) in STPTAB to describe the loader configuration.

Format:

Location	Result	Operand
label	BUILD	ML,(LE@MLT),(LDR=*loader*,ID1=*id*,ID2=*id*,
		ID3=*id*,ID4=*id*,STAT=*stat*,TYPE=*type*,CP=*cp*,
		MT=*mt*,SCRL=*scrl*,DLY=*dly*,ALTQ=*altq*,
		VSNV=*vsnv*,MMO=*mmo*,SVNL=*svnl*,SVAL=*sval*,
		SVSL=*svsl*)

Argument:

label A label referenced by ALTQ=*label* in another MLT entry

Parameters:

LDR=*loader*
ASCII loader name, left-justified, zero-filled

ID1=*id* ASCII server ID of the primary station used to communicate with the loader

ID2=*id* ASCII server ID of the second station used to communicate with the loader

ID3=*id* ASCII server ID of the third station used to communicate with the loader

ID4=*id* ASCII server ID of the fourth station used to communicate with the loader

STAT=*stat*

Initial loader status. The codes are as follows:

ML$DWN	Loader is down
ML$UP	Loader is up
ML$MAN	Loader is in manual mode (up)
ML$AUTO	Loader is in unattended mode (up)

TYPE=*type*

Loader type. The types are as follows:

ML$TMAN	Manual loader
ML$TIBMS	IBM scratch loader
ML$T4400	StorageTek 4400
ML$TM860	Master M860

CP=*cp* Communication path to the server. The values are as follows:

ML$PNS	No server/path
ML$PIN	Internal path to server
ML$PSCP	SCP path to server
ML$PSL	SUPERLINK path to server

MT=*mt* SCP message type/class. The values are as follows:

ML$SNM	No SCP message class
ML$S100	Type 3 class 100 messages
ML$S400	Type 3 class 340 messages
ML$S1	Type 1 messages

SCRL=*scrl*

Label type mask describing the type of scratch tapes supported by the autoloader. The values are as follows:

ML$SNL	Allow nonlabeled scratch tapes
ML$SAL	Allow ANSI-labeled scratch tapes
ML$SSL	Allow IBM-labeled scratch tapes
ML$SANY	Allow all label types

DLY=*dly* Maximum wait time in seconds for an available device. The possible values are as follows:

1 to 65534 seconds

ML$DMAX Infinite delay

ALTQ=*altq*
Alternate queue. This enables multiple loaders to share the same loader queue.

VSNV=*vsnv*
Verify VSN with operator for NL mounts. The two values are as follows:
0 Do not verify
1 Verify

MMO=*mmo* Echo mount messages to the master operator. The two values are as follows:
0 Do not echo
1 Echo

SVNL=*svnl*
Scratch nonlabeled volume serial name. Zero to use the system default value.

SVAL=*sval*
Scratch ANSI-labeled volume serial name. Zero to use the system default value.

SVSL=*svsl*
Scratch IBM-labeled volume serial name. Zero to use the system default value.

Examples:

1. MLT entry for a MANUAL loader (i.e., operators):

```
MLT0      BUILD       ML,(LE@MLT), (LDR='MANUAL'L,ID1='AP',STAT=ML$UP,
                      TYPE=ML$TMAN,CP=ML$PSCP,MT=ML$S100,SCRL=ML$SANY,
                      DLY=D'600,VSNV=1,MMO=1)
```

2. MLT entry for an IBM scratch loader:

```
MLT1      BUILD       ML,(LE@MLT), (LDR='SCRATCH'L,ID1='AP',
                                   STAT=ML$UP,TYPE=ML$TIBMS,
                                   CP=ML$PSCP,MT=ML$S100,
                                   SCRL=ML$SANY,DLY=D'600,
                                   VSNV=1,MMO=1,ALTQ=MLT0)
```

3. MLT entry for a StorageTek 4400 loader:

```
MLT1      BUILD       ML,(LE@MLT), (LDR='ACS-0'L,STAT=ML$UP,
                                   ID1='V3',ID2='BM',ID3='DH',
                                   TYPE=ML$T4400,CP=ML$PSCP,
                                   MT=ML$S400,SCRL=ML$SSL,
                                   SVSL='PRIVAT',DLY=D'600,VSNV=0,
                                   MMO=1)
```

2.11.11 MEDIA EXPORT TABLE (MLE)

COS uses this table to determine if a volume can be shared between loaders. Each word entry corresponds to an MLT ordinal. Each bit of the MLE word also corresponds to an MLT ordinal.

Setting bit N in word entry M indicates that to move a volume from the domain of loader M (ordinal m of the MLT) to the domain of loader N requires a controlled export/import. If the bit is clear, a mount may simply be issued to the device without performing an export/import.

Example:

```
L$OPR     =     LDR00                Operators
L$IBM     =     LDR01                IBM scratch loader
L$ACS0    =     LDR02                STK 4400 (ACS0)
L$ACS1    =     LDR03                STK 4400 (ACS1)

          CON   'MLE'L+NE@MLT
B@MLE     BSS   0
          CON   L$ACS0+L$ACS1        Manual volumes
          CON   L$ACS0+L$ACS1        IBM volumes
          CON   L$OPR+L$IBM+L$ACS1   ACS0 volumes
          CON   L$OPR+L$IBM+L$ACS0   ACS1 volumes
```

The table is defined so that moving volumes that are under the domain of the loader type OPERATORS (word 0 of the MLE) to the domain of ACS0 or ACS1 (both STK 4400 autoloaders) requires a controlled import. Moving a volume from the OPERATORS domain to the domain of the IBM scratch loader does not require a controlled export/import.

For volumes in the domain of ACS0 or ACS1, controlled exports are required to move the volume into the domain of any other loader type.

2.11.12 MEDIA MOVEMENT TABLE (MLM)

Once COS has used the Media Export Table (MLE) and determined that a volume must be moved in a controlled fashion, it uses the MLM to determine if a volume can be moved from the domain of one loader to the domain of another in order to resolve a device shortage. Setting bit N of word entry M of the MLE indicates that the volume can be moved from the domain of loader M (ordinal M of the MLT) to the domain of loader N. If the bit is clear, the operator will be informed of the device shortage but will not have the option of moving the volume.

Example:

```
L$OPR    =      LDR00            Operators
L$IBM    =      LDR01            IBM scratch loader
L$ACS0   =      LDR02            STK 4400 (ACS0)
L$ACS1   =      LDR03            STK 4400 (ACS1)

         CON    'MLE'L+NE@MLT
B@MLE    BSS    0
         CON    L$ACS0           Manual volumes
         CON    L$ACS0           IBM volumes
         CON    L$OPR+L$IBM      ACS0 volumes
         CON    0                ACS1 volumes
```

The table is defined so that Manual volumes can move to the domain of the STK 4400, ACS0, but not to the domain of the IBM scratch loader or the STK 4400, ACS1. ACS1 cannot trade volumes with any other loader.

The following is a summary of MLE and MLM characteristics:

1. If word M, bit N of the MLE is 0:
   A volume can be moved from the domain of loader M to the domain of loader N by simply issuing a mount to loader N's device.

2. If word M, bit N of the MLE is 1, and word M, bit N of the MLM is 1:
   A volume can be moved from the domain of loader M to the domain of loader N through a controlled export/import.

3. If word M, bit N of the MLE is 1, and word M, bit N of the MLM is 0:
   A volume cannot be moved from the domain of loader M to the domain of loader N.

2.12 CONFIGURATION EXAMPLES

Figure 2-1 shows COS table entries representing a configuration of six disk units on a system without an IOS. The system has four DD-19 Disk Storage Units (DSUs) attached to a disk control unit (DCU) on software channel pair 2 (hardware channels 4 and 5) and two DD-29 DSUs attached to another DCU on software channel pair 3 (hardware channels 6 and 7).

```
          INSTALLATION PARAMETERS        (DECK=COSI@P)
  I@DD      =         D'6           Total number of logical disk units
  I@IOPICH =          0             I/O Subsystem input channel number

          EQUIPMENT TABLE (EQT)          (DECK=STPTAB)

  EQT20     EQT       LDV='DD-19-20',CHN=2,UNT=0,DT=DD19,MBT=1,MSD=1
  EQT30     EQT       LDV='DD-29-30',CHN=3,UNT=0,DT=DD29
  EQT21     EQT       LDV='DD-19-21',CHN=2,UNT=1,DT=DD19,MBT=1
  EQT31     EQT       LDV='DD-29-31',CHN=3,UNT=1,DT=DD29
  EQT22     EQT       LDV='DD-19-22',CHN=2,UNT=2,DT=DD19,MBT=1
  EQT23     EQT       LDV='DD-19-23',CHN=2,UNT=3,DT=DD19,MBT=1
          EQTEND
```

Figure 2-1. Sample Disk Configuration without an IOS

Figure 2-2 shows a system with an IOS. The system has four DD-19 DSUs connected to IOP-1 (the Buffer I/O Processor (BIOP)) and two DD-29 DSUs connected to IOP-2 (a Disk I/O Processor (DIOP)). The DD-19 DSUs on IOP-1 channels 21 and 23 are striped (note the GRP=1 entry).

```
*         INSTALLATION PARAMETERS          (DECK=COSI@P)
I@DD      =         D'7           Total number of logical disk units
I@IOPICH =          2             I/O Subsystem input channel number


*         EQUIPMENT TABLE (EQT)          (DECK=STPTAB)

EQT120    EQT       LDV='DD-A1-20',IOP=1,CHN=20,DT=DD19,MSD=1
EQT121    EQT       LDV='DD-A1-21',IOP=1,CHN=21,DT=DD19,GRP=1
EQT122    EQT       LDV='DD-A1-22',IOP=1,CHN=22,DT=DD19
EQT123    EQT       LDV='DD-A1-23',IOP=1,CHN=23,DT=DD19,GRP=1
EQT220    EQT       LDV='DD-A2-20',IOP=2,CHN=20,DT=DD29
EQT221    EQT       LDV='DD-A2-21',IOP=2,CHN=21,DT=DD29
STRIPE1   EQT       LDV='STRIPE-1',NA=1,IOP=0,CHN=20
          EQTEND

*         IOS CONFIGURATION

*         IOP-1 CHANNEL TABLE          (DECK AMAP)

          CHANNEL   (20),TY=D1              .DD-A1-20
          CHANNEL   (21),TY=D1S             .DD-A1-21 (striped)
          CHANNEL   (22),TY=D1              .DD-A1-22
          CHANNEL   (23),TY=D1S             .DD-A1-23 (striped)
          CHANNEL   (24,37),TY=(EM,D'14)    .EMPTY

*         IOP-2 CHANNEL TABLE          (DECK AMAP)

          CHANNEL   (20),TY=D2              .DD-A2-20
          CHANNEL   (21),TY=D2              .DD-A2-21
          CHANNEL   (22,37),TY=(EM,D'12)    .EMPTY

*         LOGICAL DEVICE TABLE         (DECK=AMAP)

          LDEV      TYPE=DSK,CHN=20,HDS=10,SCS=18,UNS=(121,123)
          LDEV      END
```

Figure 2-2. Sample Disk Configuration with an IOS

Figure 2-3 shows configuring of 262,144 words of Buffer Memory for COS BMR datasets. In the EQT, the RBN flag is set so datasets are not automatically assigned to the logical device. The processor number IOP must be 0 and the channel number (CHN) must match the channel defined in the LDEV Table in IOS AMAP. Buffer Memory is not configured as a controlled device in this example.

```
*         INSTALLATION PARAMETERS       (DECK=CONFIG@P)

C@NSEBM  =          D'512     Number of sectors configured for Buffer
                              Memory

*         EQUIPMENT TABLE (EQT)         (DECK=STPTAB)

EQTBMR   EQT        LDV='BMR-0-20',CHN=20,DT=EBM,IOP=0,RBN=1

*         IOS CONFIGURATION

*         INSTALLATION PARAMETERS       (DECK=AT)

BMR@SIZ  EQUALS  4     .Number of 200000₈ word segments for BMR

*         LOGICAL DEVICE TABLE          (DECK=AMAP)

         LDEV    TYPE=BMR,CHN=20,BLKS=BMR@SIZ
         LDEV    END
```

Figure 2-3. Sample EQT, Configuring One-fourth Million Words of Buffer Memory for BMR Datasets

Figure 2-4 shows a sample disk configuration with DD-39 and DD-49 DSUs on a system with an IOS. The system has two DD-49 DSUs connected to IOP-1 channels 20 and 21, and six DD-39 DSUs connected to IOP-2 channels 20 and 21. Each IOP-2 channel supports three DD-39 disk units.

```
*        INSTALLATION PARAMETERS       (DECK=COSI@P)

  I@DD       =    D'8                 Total number of logical disk units
  I@IOPICH   =    D'8                 I/O Subsystem input channel number

*        EQUIPMENT TABLE (EQT)         (DECK=STPTAB)

EQT120A  EQT        LDV='49-1-20A',IOP=1,CHN=20,DT=DD49,MSD=1
EQT121A  EQT        LDV='49-1-21A',IOP=1,CHN=21,DT=DD49
EQT220A  EQT        LDV='39-2-20A',IOP=2,CHN=20,DT=DD39,UNT=0
EQT221A  EQT        LDV='39-2-21A',IOP=2,CHN=21,DT=DD39,UNT=0
EQT220B  EQT        LDV='39-2-20B',IOP=2,CHN=20,DT=DD39,UNT=1
EQT221B  EQT        LDV='39-2-21B',IOP=2,CHN=21,DT=DD39,UNT=1
EQT220C  EQT        LDV='39-2-20C',IOP=2,CHN=20,DT=DD39,UNT=2
EQT221C  EQT        LDV='39-2-21C',IOP=2,CHN=21,DT=DD39,UNT=2
         EQTEND

*        IOS CONFIGURATION

*        IOP-1 CHANNEL TABLE          (DECK=AMAP)

         CHANNEL    (20),TY=D4              .49-1-20A
         CHANNEL    (21),TY=D4              .49-1-21A
         CHANNEL    (22,37),TY=(EM,D'14) .EMPTY

*        IOP-2 CHANNEL TABLE          (DECK AMAP)

         CHANNEL    (20),TY=D3              .39-1-20(A,B,C)
         CHANNEL    (21),TY=D3              .39-1-21(A,B,C)
         CHANNEL    (22,37),TY=(EM,D'14) .EMPTY
```

Figure 2-4. Sample Disk Configuration with DD-39 DSUs and DD-49 DSUs

Figure 2-5 shows the configuration of three DD-39 striped devices with each striped device composed of two DD-39 units residing on IOP-1; and one DD-29 device residing on IOP-1.

```
*    INSTALLATION PARAMETERS       (DECK=COSI@P)

I@DD        =   D'10      Total number of logical disk units
I@IOPICH    =   D'8       IOS input channel number


*    EQUIPMENT TABLE (EQT)         (DECK STPTAB)

EQT120     EQT       LDV='29-1-20A',IOP=1,DT=DD29,CHN=20,MSD=1
EQT121A    EQT       LDV='39-1-21A',IOP=1,DT=DD39,CHN=21,UNT=0,GRP=1
EQT121B    EQT       LDV='39-1-21B',IOP=1,DT=DD39,CHN=21,UNT=1,GRP=2
EQT121C    EQT       LDV='39-1-21C',IOP=1,DT=DD39,CHN=21,UNT=2,GRP=3
EQT122A    EQT       LDV='39-1-22A',IOP=1,DT=DD39,CHN=22,UNT=0,GRP=1
EQT122B    EQT       LDV='39-1-22B',IOP=1,DT=DD39,CHN=22,UNT=1,GRP=2
EQT122C    EQT       LDV='39-1-22C',IOP=1,DT=DD39,CHN=22,UNT=2,GRP=3
STRIPE1    EQT       LDV='STRIPE-1',IOP=0,DT=DD39,CHN=30,NA=1
STRIPE2    EQT       LDV='STRIPE-2',IOP=0,DT=DD39,CHN=31,NA=1
STRIPE3    EQT       LDV='STRIPE-3',IOP=0,DT=DD39,CHN=32,NA=1
           EQTEND

*    IOS CONFIGURATION

*    IOP-1 CHANNEL TABLE         (DECK=AMAP)

     CHANNEL    (20),TY=D2              .29-1-20A
     CHANNEL    (21),TY=D3S             .39-1-21(A,B,C)  STRIPED
     CHANNEL    (22),TY=D3S             .39-1-22(A,B,C)  STRIPED
     CHANNEL    (23,37),TY=(EM,D'13)    .EMPTY

*    LOGICAL DEVICE TABLE          (DECK=AMAP)

     LDEV        TYPE=DSK,CHN=30,HDS=5,SCS=24,UNS=(121,122)
     LDEV        TYPE=DSK,CHN=31,HDS=5,SCS=24,UNS=(121,122)
     LDEV        TYPE=DSK,CHN=32,HDS=5,SCS=24,UNS=(121,122)
     LDEV        END
```

Figure 2-5. Configuring Three DD-39 Striped Devices

Figure 2-6 shows entries configuring a 32-million-word SSD as a Controlled Device.

```
*          EQUIPMENT TABLE (EQT)         (DECK=STPTAB)

EQTSSD     EQT        LDV='SSD-0-20',DT=SD32,RBN=1,GRN='SSD'L

*          GENERIC RESOURCE TABLE (GRT)

GRT01      BUILD      GR,(LE@GRT ),(NM='SSD'L)
```

Figure 2-6. Sample EQT and GRT Configuring a 32-million-word SSD

Figure 2-7 shows the table entries configuring an 8-million-word SSD.

```
*          EQUIPMENT TABLE (EQT)         (DECK=STPTAB)

EQTSSD     EQT        LDV='SSD-0-20',DT=SD8,RBN=1
```

Figure 2-7. Sample EQT Configuring an 8-million-word SSD

Figure 2-8 shows an example of splitting one physical device into multiple logical devices. In this example, a 128-million-word SSD has been evenly broken into three separate logical devices; a scratch device, a controlled device, and a request-by-name device.

```
*           EQUIPMENT TABLE  (EQT)

EQTSSD      EQT  LDV='SSD-SCRT',DT=SD128,SCR=1,STK=0,NTK=5252
EQTSS2      EQT  LDV='SSD-CNTL',DT=SD128,RBN=1,GRN='SSD'L,STK=5252,__
,_____________________NTK=5253
EQTSS3      EQT  LDV='SSD-RQST',DT=SD128,RBN=1,STK=12525,NTK=5253

*           GENERIC RESOURCE TABLE (GRT)

GRT01       BUILD GR,(LE@GRT ),(NM='SSD'L)
```

Figure 2-8. Splitting One Physical Device into Multiple Logical Devices

2.12.1 CONFIG MACRO EXAMPLES

The four COSDEF installation parameters required to define an on-line tape configuration follow. These parameters show a sample configuration containing a 3-by-5 bank of tapes and a 1-by-2 bank of tapes. The designation 3-by-5 means that a group of five tape devices can be accessed by any of three control units, one control unit per XIOP block multiplexer channel. In this example, both tape banks are string shared with a front-end computer (only one configuration table change is required to support this capability, namely setting the STS=DOWN parameter).

Example:

```
I@TS=1          Enable tape I/O
I@NETDT=7       Number of TDT entries (one for each device)
I@TNTB=2        Number of tape banks in the system
I@SFEN=1        Servicing front-end catalog is enabled
```

The following lists of procedures are examples of the CONFIG macro:

1. Configure a 3-by-5 tape configuration (three control units that can access five tape transports):

```
CNT0   CONFIG  DVN=530,BANK=0,UN=0,DID=0,ICH1=20:3,CT=SELECT,
,_____________ICH2=21:4,ICH3=22:5,DT=DT@GCPE,CTL=TDT0,GDN=*TAPE
CNT1   CONFIG  DVN=531,BANK=0,UN=1,DID=1,ICH1=20:3,CT=SELECT,
,_____________ICH2=21:4,ICH3=22:5,DT=DT@GCPE,CTL=TDT1,GDN=*TAPE
CNT2   CONFIG  DVN=538,BANK=0,UN=2,DID=8,ICH1=20:3,CT=SELECT,
,_____________ICH2=21:4,ICH3=22:5,DT=DT@GCPE,CTL=TDT2,GDN=*TAPE
CNT3   CONFIG  DVN=53A,BANK=0,UN=3,DID=A,ICH1=20:3,CT=SELECT,
,_____________ICH2=21:4,ICH3=22:5,DT=DT@GCPE,CTL=TDT3,GDN=*TAPE
CNT4   CONFIG  DVN=539 BANK=0,UN=4,DID=9,ICH1=20:3,CT=SELECT,
,_____________ICH2=21:4,ICH3=22:5,DT=DT@GCPE,CTL=TDT4,GDN=*TAPE
```

2. Configure a 1-by-2 with one drive initially connected but not available:

```
CNT0   CONFIG  DVN=TAPE1,BANK=0,UN=0,DID=4,ICH1=20:A,CT=SELECT,
,_____________DT=DT@GCPE,CTL=TDT0,STS=DOWN,GDN=*TAPE
CNT1   CONFIG  DVN=TAPE2,BANK=0,UN=1,DID=0,ICH1=20:A,CT=SELECT,
,_____________DT=DT@GCPE,CTL=TDT1,GDN=*TAPE
```

3. Configure a 2-bank system with a 2-by-2 and a 1-by-3 configuration:

```
TAG1   CONFIG  DVN=TAPE1,BANK=0,UN=0,DID=0,ICH1=20:C,CT=SELECT,
,_____________ICH2=22:4,DT=DT@GCPE,CTL=TAPE1,GDN=*TAPE
TAG2   CONFIG  DVN=TAPE2,BANK=0,UN=1,DID=1,ICH1=20:C,CT=SELECT,
,_____________ICH2=22:4,DT=DT@GCPE,CTL=TAPE2,GDN=*TAPE
TAG3   CONFIG  DVN=TAPE3,BANK=1,UN=2,DID=0,ICH1=21:2,CT=SELECT,
,_____________DT=DT@GCPE,CTL=TAPE3,GDN=*TAPE
TAG4   CONFIG  DVN=TAPE4,BANK=1,UN=3,DID=1,ICH1=21:2,CT=SELECT,
,_____________DT=DT@GCPE,CTL=TAPE4,GDN=*TAPE
TAG5   CONFIG  DVN=TAPE5,BANK=1,UN=4,DID=2,ICH1=21:2,CT=SELECT,
,_____________DT=DT@GCPE,CTL=TAPE5,GDN=*TAPE
```

4. Configure a drive with two control unit accesses, with one unit temporarily inoperative:

```
CNTA   CONFIG  DVN=TAPE1,BANK=4,UN=12,DID=A,ICH1=20:C:OFF,CT=SELECT,
,_____________ICH2=21:D,DT=DT@GCPE,CTL=TAPEA,GDN=*TAPE
```

5. Configure a drive with two control unit accesses, with one IOP channel temporarily inoperative:

```
CNTA   CONFIG  DVN=TAPE1,BANK=5,UN=20,DID=C,ICH1=20:D,CT=SELECT,
,______________ICH2=22:OFF:A,DT=DT@GCPE,CTL=TDT1,GDN=*TAPE
```

6. Configure a CPU entry to allow operator CPU UP/DOWN CONFIG commands:

```
CONFIG  DVN=XMP2CPU,DT=DT@CPU
```

7. Configure a 2-by-2 system with 3480 tapes with both drives down:

```
CONFIG  DVN=200,GDN=*CART,UN=4,DT=DT@3480,STS=DOWN,CTL=TDT04
,________CT=DTSTR,BANK=1,DID=0,IOP=3,ICH1=24:4,ICH2=25:5
CONFIG  DVN=201,GDN=*CART,UN=5,DT=DT@3480,STS=DOWN,CTL=TDTOS
,________CT=DTSTR,BANK=1,DID=1,IOP=3,ICH1=24:4,ICH2=25:5
```

2.12.2 GENERIC RESOURCE TABLE (GRT) EXAMPLE

Generic resource names are communicated to COS with the GRT. The GRT is located in the System Task Processor (STP) tables. The table contains one entry for each generic resource defined. The generic resource names declared in the EQT and in the CNT must match those declared in the GRT. The number of GRT entries configured must be reflected by the installation parameter I@NGRN. Figure 2-9 shows an example of a GRT definition.

```
GRT00    BUILD    GR,(LE@GRT ),(GRN='1600'L)
GRT01    BUILD    GR,(LE@GRT ),(GRN='SSD'L)
GRT02    BUILD    GR,(LE@GRT ),(GRN='BMR'L)
GRT03    BUILD    GR,(LE@GRT ),(GRN='6250'L)
GRT04    BUILD    GR,(LE@GRT ),(GRN='*CART'L)
GRTOS    BUILD    GR,(LE@GRT),(GRN='ISPI1'L,TOT=1)
```

Figure 2-9. Generic Resource Name Definition

2.12.3 IOP INFORMATION TABLE EXAMPLES

The following examples are IOP Information Tables for the IOS. In this example, a total of 4000_8 sectors (1 million words) of Buffer Memory are reserved for IOS system use. Thus, the IOS installation parameter IOS@SIZ should be set to 20_8 ($4000_8/200_8=20_8$). Any additional space in Buffer Memory could be used for Buffer Memory Resident datasets.

```
                *       IOP-0 Information
    AAPT *
                400     Total number of 512-word sectors of Buffer Memory
                        for this IOP
                D'8     Local 512-word data buffers
                D'48    Buffer Memory software stacks
                32000   Local Memory overlay space
                D'48    Local Memory message packets
                1       Amount of 65K Local Memory parcels
                AACH    Channel configuration
                TTL0    Deadstart title address
                0       IOP options
                CBF0    Deadstart operator commands start
                CBF0L   Deadstart operator commands end

                *       IOP-1 Information
ABPT *
                1200    Total number of 512-word sectors of Buffer Memory
                        for this IOP
                D'20    Local 512-word data buffers
                D'10    Buffer Memory software stacks
                14000   Local Memory overlay space
                D'64    Local Memory message packets
                1       Amount of 65K Local Memory parcels
                ABCH    Channel configuration
                TTL1    Deadstart title address
                0       IOP options
                CBF1    Deadstart operator commands start
                CBF1L   Deadstart operator commands end

                *       IOP-2 Information
ACPT *
                1000    Total number of 512-word sectors of Buffer Memory
                        for this IOP
                D'20    Local 512-word disk buffers
                D'10    Buffer Memory software stacks
                14000   Local Memory overlay space
                D'64    Local Memory message packets
                1       Amount of 65K Local Memory parcels
                ACCH    Channel configuration
                TTL2    Deadstart title address
                0       IOP options
                CBF2    Deadstart operator commands start
                CBF2L   Deadstart operator commands end
```

```
            *       IOP-3 Information
ADPT *
            1200    Total number of 512-word sectors of Buffer Memory for
                    this IOP
            D'17    Local 512-word disk buffers
            D'32    Buffer Memory software stacks
            24000   Local Memory overlay space
            D'32    Local Memory message packets
            1       Amount of 65K Local Memory parcels
            ADCH    Channel configuration
            TTL3    Deadstart title address
            0       IOP options
            CBF3    Deadstart operator commands start
            CBF3L   Deadstart operator commands end
```

2.12.4 SITE-SPECIFIC TARGET MACHINE NAME CONFIGURATION EXAMPLE

This example shows how a site defines specific characteristics for a machine name. For example, a site wants to add a primary machine type name of ALPHA to designate a 2-Mword CRAY-1 M computer system. The BUILD macro constructs a machine characteristics table. Figure 2-10 shows the mod to implement this change.

The CON specifies the new name for the machine type being defined. The first two parameters of the BUILD macro must be MC,SZ and cause the macro to build a full machine characteristics table.

```
*IDENT ADDPMT,DC=GETPMC
*I GETPMC.101
         CON          'ALPHA'L
ALPHA    BUILD        MC,SZ,(PMT='CRAY-1'L,BANK=D'8,NCPU=1,IBSZ=C'16,_____
,________MSZ=MC2MEG,MSP=D'13,CLK=D'12500,NCL=0,BBSY=D'8,___________________
,________EMA=FALSE,CIGS=FALSE,VPOP=TRUE,PC=TRUE,RDVL=FALSE,VRCR=TRUE,_____
,________AVL=FALSE,HPM=FALSE,BDM=FALSE,STR=FALSE)
```

Figure 2-10. Configuring Machine Characteristics Table for a 2-Mword CRAY-1 M Computer System

3. COS INSTALLATION

This section describes the on-site installation of COS software. The first subsection outlines the two different types of software releases. The next subsections describe software installation differences at sites with new and existing computer systems. The remaining subsections deal with COS security, permanent dataset privacy, and archiving.

3.1 SOFTWARE RELEASES

The two types of COS software releases are the feature release and the revision release. Both are concerned with the same set of software, but their purpose and content differ. In each release, customers receive documentation describing the release, along with the appropriate order forms necessary for ordering the release media.

3.1.1 FEATURE RELEASE

Cray Research, Inc. (CRI) periodically releases the complete set of COS software, including new features and problem fixes (revisions). A release includes the software, the software documentation, a Release Preview/Release Notice.

The software is released in the form of a set of tapes. The documentation describing the software can be manuals, change packets, or both.

The Release Preview is a document providing information about a feature release. The Release Preview are sent to the Cray field support and customers before each feature release.

The Release Notices contains descriptions of new features, software enhancements, and end-user impact.

within each category a brief description is provided, followed by the advantages, end-user impact, systems impact, operations impact, field engineer impact, non-COS impact, dependencies, and code changes. An appendix lists the publications affected by this release and an order form is provided to request the appropriate release package for the site.

A System Installation Bulletin describes the specific contents of the release tapes and provides detailed installation procedures for the release.

The tapes include one copy of the following:

- Released program libraries (PLs) containing all the source code in UPDATE format
- Released program binaries for a minimally configured system
- Modifications that were applied to the previously released base PLs to generate the new PLs
- MODSTATUS that describes all of the modifications
- Regression Test Base (RTB) to run against the system

The tape format is compatible with either an I/O Subsystem (IOS) or a Data General ECLIPSE performing maintenance control functions. There is also an optional tape containing PLs and Mods in 6250 bpi on-line tape format.

Each site receives two copies of the Release Notice, one for the CRI analyst-in-charge (AIC) and one for the customer. The AIC is responsible for delivering the Notice Letter to the customer. The Release Notice is distributed upon release of the software.

Interface software services are available for the following front-end operating systems and mainframes:

Operating System	Mainframe
NOS	CDC mainframes
NOS/BE	CDC mainframes
MVS	IBM-compatible mainframes
RDOS	Data General ECLIPSE minicomputers
VAX/VMS	DEC mainframes
VM	IBM-compatible mainframes

Interface software services are also available for many front-end mainframes running the UNIX operating system.

3.1.2 REVISION RELEASE

Several times between feature releases, revision releases are made to fix problems found in the feature release. These revision releases contain the updated program libraries (PLs), as well as the mods applied to the feature release to create these new PLs. A Release Letter also accompanies the revision release.

The Release Letter is sent to Cray Field Support and customers upon release of the software. Order forms are provided so site may request the appropriate package for the release.

3.2 OVERVIEW OF SOFTWARE INSTALLATION ON A NEW CRAY COMPUTER SYSTEM

General assumptions to be made about how to install CRI software on a Cray computer system depend on whether the system has a Data General ECLIPSE or a CRI IOS as the maintenance control unit (MCU).

3.2.1 INSTALLATION WITH THE DATA GENERAL ECLIPSE AS THE MCU

With the Data General ECLIPSE S/200 or S/230 as the MCU, the following assumptions are made:

- The CRI disk controller 0 is connected to CPU channels 4 and 5.
- A DD-19 or a DD-29 disk is connected to controller 0, unit 0.
- The mainframe has at least 1 Mword of Central Memory.

The ECLIPSE station runs under the control of Data General's RDOS operating system. The DG local release includes the binaries necessary for running the DG station.

The Data General (DG) station software executes under RDOS. Source code and manuals are part of the DG station release materials. The Data General Station (DGS) Operator's Guide describes how to use the Data General station. The Data General Station (DGS) Internal Reference Manual describes internal characteristics of the DG station.

Two ECLIPSE disk packs are delivered with the system: the install pack and the operations pack. Use the install pack for the install Startup option; it contains all standard COS software, either as released or as modified for the specific site. Use the operations pack for all normal operations of the ECLIPSE, which include job entry station, operator station, and MCU for deadstart and restart options. The only COS programs that must be on the operations pack are COS and DDC.

The Release Notice accompanying the release materials outlines the specific procedures for installing the software.

3.2.2 INSTALLATION WITH THE IOS AS THE MCU

With the IOS configured as the MCU, the following assumptions are made:

- Two I/O Processors (IOPs), the Master IOP (MIOP) and the Buffer IOP (BIOP), are present.
- The MIOP is connected to CPU channels 2 and 3 on a CRAY-1 S or CRAY-1 M computer system and to CPU channels 8 and 9 on a CRAY X-MP computer system.
- A disk is connected to BIOP channel 20. This is a DD-29 disk on a CRAY-1 computer system, and either a DD-29, DD-39, DD-40, or DD-49 on a CRAY X-MP computer system.
- At least one-half million words of Buffer Memory are present, and one million words of Central Memory.

This section does not assume familiarity with COS or with IOS operator commands. Problems can occur, however, and knowledge of both IOS station commands and COS are required to isolate and resolve the problems. See the I/O Subsystem (IOS) Operator's Guide for COS for descriptions of the commands.

3.2.3 SYSTEM DIRECTORY

The System Directory (SDR) is a table in STP that must be initialized after an Install Startup option. Entries are added to the SDR by specifying the ENTER parameter on an ACCESS or ACQUIRE control statement. A job to enter all the standard software into the SDR is included in GENPL as deck JSYSDIR, and is described in the Release Notice accompanying this manual.

Entries are added or changed at any time by running a job with the ENTER parameter on the ACCESS or ACQUIRE control statement. Once a dataset is in the SDR, user jobs that want to load and execute the dataset need not use an ACCESS or ACQUIRE statement before using the name of the dataset as a control statement verb.

Formats:

```
ACCESS, ... ,ENTER, ... .
ACQUIRE, ... ,ENTER, ... .
```

During a Restart or Deadstart, the SDR is recovered unless an *SDR directive is in the startup parameter file. If Startup is unable to recover the SDR, a message is issued to the system log and to the operator, and the operating system initialization is abnormally terminated. If an *SDR directive is in the startup parameter file, the SDR is not recovered.

NOTE

If the ED parameter is not specified on the ACCESS or ACQUIRE statement, then the highest edition is placed in the SDR, both at ACCESS/ACQUIRE time and SDR recovery time.

3.3 INSTALLING SOFTWARE ON AN EXISTING CRAY X-MP OR CRAY-1 COMPUTER SYSTEM

Any site running a system two or more releases older than the current feature release may encounter problems implementing the new release. CRI tests the procedure of upgrading for consecutive releases only; CRI does not test upgrading to the new release from releases other than the previous release (for example, upgrading from COS version 1.15 to 1.17 is not tested). The CRI representative and the customer must determine the best way to implement the new release.

Installing new software on an existing CRAY X-MP or CRAY-1 computer system differs in two important ways from installing new software on a new CRAY X-MP or CRAY-1 computer system. First, an existing system does not require an install Startup option; rather, the new release can be built upon the existing base of permanent datasets. Second, an existing system has active users not wanting to lose time from possible problems associated with installation or with the new software itself. The Release Notice that accompanies the release materials contains the specific procedures required to make a smooth and controlled transition from one release level to the next.

3.4 SECURITY ENVIRONMENT

Transition to a full security environment is facilitated by defining four levels of security: QUIET, WARN, ABORT, and CRYPT. The following is a list describing the levels of security including the parameters that pertain to the levels (see appendix A for descriptions of the parameters):

Security	Description
QUIET	All privilege checks are in place and processed. Illegal requests are processed without notification, as if there were no checks being made. (I@SLVL=-1)
WARN	All privilege checks are in place and processed. Illegal requests are honored, but the user is warned that an illegal request has been made. All security tracking messages are entered into the system log dataset. (I@SLVL=0)
ABORT	All privilege checks are in place and processed. The user job is aborted if an illegal request is received by COS. All security tracking messages are entered into the system log dataset. (I@SLVL=1)
CRYPT	Same as ABORT mode with the added feature of password encryption. (I@CRYPT=1)

NOTE

In all four modes, system access is enforced when the ACCOUNT statement is processed.

3.4.1 IMPLEMENTING THE SECURITY MECHANISM

Follow these steps when implementing the security mechanism described in this manual:

1. Determine the level of security desired (see previous subsection). Generate a new COSTXT/COSDEF and $SYSTXT/$SYSDEF with I@SLVL and I@CRYPT set to the appropriate values. Save the new COSTXT/COSDEF and $SYSTXT/SYSDEF.

2. Determine the methods of user validation. The requires the $VALIDATION dataset using the PRVDEF utility and the $ACCTN dataset using the ACCTDEF utility. The user limits method requires the RESOURCEDATASET using the RDGEN utility. The ACCOUNT program is not required in this system.

3. Create the required validation dataset from the following:
   - $VALIDATION from PRVDEF
   - $ACCOUNT from ACCTDEF
   - $RD from RDGEN

4. Save the validation datasets with the required passwords.

5. Generate a new system, including CSP and $SYSLIB, using the new COSDEF and $SYSDEF.

6. Using the new COSDEF and $SYSDEF, create the new ACCOUNT, CHARGES, ACCTDEF, AUDIT, PDSDUMP, PDSLOAD, PRIVDEF, and any other programs requiring special privileges for the security system (see GENPL in subsection 1.1.2, Configuration of GENPL).

7. Enter *SDR in the Startup parameter file. The first job run after the system is started should create the new SDR installing the modules created in step 6.

8. Startup the new system. If the security level is ABORT or WARN, the following precautions should be taken with the validation datasets.

   - The passwords for accessing the validation dataset should be changed in ACCOUNT and CHARGES to secure these datasets.

   - If permanent dataset privacy is being used, the $VALIDATION should be resaved with the new passwords and system dataset owner ID.

The security mechanism is now in place.

The following programs, shown with their needed minimum privileges, should be saved as executive only (EXO) datasets.

Program	Minimum Security
ACCOUNT	SPRIV
AUDIT	SCRDSC
CHARGES	SCPRIV
PDSDUMP	SCRDSC, SCDTIM, SCUPDD (and SCSPOL, if a general user can dump spooled datasets)
PDSLOAD	SCRDSC, SCLUSR (and SCSPOL, if a general user can load spooled datasets)
RDVAL	SCRDSC or run under by USGR with this privilege

3.4.2 CRAY COMPUTER SYSTEM SECURITY HINTS

ABORT is the default security mode and is set by UPDATE deck USERI@P:

```
I@SLVL  = 1
I@CRYPT = 0 or 1
```

WARN security mode can be initiated by making the following modifications to UPDATE deck USERI@P:

```
I@SLVL  = 0
I@CRYPT = 0 or 1
```

ABORT is the same as WARN with the following exceptions:

- A job step is aborted upon a violation.

- The job is aborted if the number of allowed security violations is exceeded.

CRYPT can be initiated in either WARN mode or ABORT mode by setting I@CRYPT=1.

Processing the ACCOUNT statement determines if a user can obtain access to the system. Until ACCOUNT is entered into the SDR, any user is permitted access to the system and given all system privileges. $VALIDATION must be generated using PRVDEF (see Operational Aids Reference Manual, SM-0044) before ACCOUNT is entered into the SDR. If the validation datasets are destroyed, perform a deadstart or restart with *SDR in the parameter file (which removes ACCOUNT from the SDR) and do the following:

1. Create the validation datasets required, the $VALIDATION using PRVDEF, the $ACCT dataset using ACCTDEF, or the $RD (Resource Dataset) using RDGEN.

2. Run the SDR job to enter ACCOUNT into SDR. When you deadstart or restart with *SDR in the parameter file, you must run the system directory job to enter all system datasets into the SDR.

3.4.3 CHANGES MADE BY THE USER

After the ACCOUNT module has been entered into the SDR, a user must supply the ACCOUNT control statement to make any changes to the system, run programs, or access datasets.

The validation datasets must contain a matching user number and password entry for a user to gain access to the system in WARN mode. Normally, the user must provide a valid user number and user password on the ACCOUNT statement. The validation datasets must also contain a matching account number, because accounting is mandatory with security.

If the site wants to minimize the impact on the user, the user account number could be entered as the user number and password using the NO PASSWORD NECESSARY option available in the validation dataset entry for each user. This option allows the user to gain access to the system without making changes to his ACCOUNT statement.

NOTE

PRVDEF creates the local dataset $USER. The installation manager must save $USER as PDN=$VALIDATION with proper passwords. ACCOUNT and CHARGES access $VALIDATION with passwords of:

R=RPASS, W=WPASS, and M=MPASS, OWN=SYSTEM

ACCTDEF creates the local dataset $ACCTN. The installation manager must save $ACCTN as PDN=$ACCOUNT with the proper passwords. ACCOUNT accesses $ACCOUNT with passwords of:

R=RPASS, W=WPASS, and M=MPASS, OWN=SYSTEM

These passwords should be changed within the ACCOUNT, CHARGES, and ACCTDEF programs if the site prefers to use different passwords.

RDGEN creates the local dataset $RD. The installation manager must save $RD as PDN=RESOURCEDATASET,ID=SYSTEM OWN=SYSTEM. If this information is changed or passwords are used, then the *RDMRDS startup parameter will be required to locate the Resource Dataset for RDM. RDM has unique access to this dataset during operation of the system with RDM active.

Giving a user special privileges requires using the PRVDEF utility with the GRANT key word for a standard system, or the RDEDIT utility with the GRANT or CHANGE directive. A user can be given the privileges described in the Operational Aids Reference Manual, publication SM-0044. Error messages related to dataset security can be found in the COS Message Manual, publication SR-0039, in the message range from AB292 through AB300 and AC109 through AC120.

The following examples generate a routine to allow any user to dump datasets (loaded module must reside in the SDR for the granted privileges to take effect):

```
    LDR,MAP,NX,AB=PDSDUMP,GRANT=SCRDSC:SCUPDD,SECURE.
or  SEGLDR,CMD='MAP=PART;ABS=PDSDUMP;SECURE;GRANT=SCRDSC,SCUPDD'.
```

3.5 PERMANENT DATASET PRIVACY

Permanent dataset privacy is an optional feature of COS. This subsection presents the following information:

- Definition of permanent dataset privacy
- Implementing permanent dataset utilities
- Privacy and the permanent dataset utilities

3.5.1 DEFINITION OF PERMANENT DATASET PRIVACY

Permanent dataset privacy consists of a set of access control mechanisms, all based on the concept of dataset ownership. The owner of a permanent dataset is normally the person who created it. The owner can define the user and conditions under which the dataset can be accessed. Access control is established by using one of the following mechanisms:

- Public access mode
- Permit access mode

3.5.1.1 Public access mode

Public access mode defines the types of access allowed all other users on the system. The types of public access available follow:

Access Mode	Description
None	Only the owner can access the dataset.
Execute	Other users can only execute the dataset.
Read	Permission to read and execute the dataset is given to other users.
Write	Permission to read, write, and execute on the dataset is given to other users.

Privacy Type	Description
Maintenance	Permission to read, write, execute, and perform maintenance on the dataset is given to other users.
Maintenance-only	Permission to perform maintenance on the dataset is given to other users.

You can specify read, write, and maintenance alone or in any combination. If no public access mode is explicitly given, the site-defined installation option for public access mode is used.

3.5.1.2 Permit access mode

Permit access mode allows specific users to access the dataset owner's permanent datasets. The modes of access for permit access mode are the same as those for public access modes. The combination of a specific user and an access mode is known as a permit.

The dataset owner can specify both public access mode and any number of permits for a particular dataset. Permits always take precedence over public access mode when a permitted user accesses a dataset.

3.5.1.3 Access control exceptions

Public access and permit access modes do not supplant any control words previously set. If control words were set when a dataset was saved or modified, they must still be specified to access the dataset and obtain the access mode controlled by the control words. Likewise, the Execute Only flag overrides any allowed permissions if it was set when the dataset was saved.

3.5.2 IMPLEMENTING PERMANENT DATASET PRIVACY

The system must be configured with privacy in mind to ensure a smooth transition from a nonprivate to a private environment.

First, decide on the ownership value from the two possible ownership values available in the released system:

- Account number
- User number

STPTAB defines I@PDOWN, the installation parameter that sets ownership value. The I@PDOWN parameter can have the value ACN or USN and is read by an STP common routine called GETOWN.

3.5.2.1 Ownership value

When GETOWN is given the Job Table Area (JTA) address of the caller, it returns the 15-character owner name. Because the owner value is determined by this routine alone, you can replace it with a local version. The ownership value can be something other than the Cray-supplied options.

You must also define the default system owner value used to identify datasets created by the system (including station saves). The default system owner value is defined by the values of:

I@SYOWN1 (characters 1 to 8 of owner value)

I@SYOWN2 (characters 9 to 15 of owner value)

I@SYOWN1 and I@SYOWN2 can be found in STPTAB. The default system owner value in the released system is defined as the following:

'SYSTEM'L.

3.5.2.2 Determining ownership value

The choice of ownership value is determined to some extent by other site options, specifically, mandatory accounting and the form of the COS security feature selected. If the security feature is enabled, both an account number and a user number are always available to the GETOWN routine. In this case, either of the ownership options can be selected.

If the security feature is not used but mandatory accounting is enforced, only the presence of the account number can be guaranteed. In this case, the account number is the obvious choice for the ownership value.

If neither security nor mandatory accounting is enabled, neither of the Cray-supplied ownership options can be guaranteed to be available to the GETOWN routine. Sites running this configuration are strong candidates for local versions of GETOWN, which use some other values available in the JTA.

3.5.2.3 Default public access mode

The second installation option of importance to permanent dataset privacy is the default public access mode (I@PDPAM) used when none is specified by the user. The common deck USERI@P defines I@PDPAM. In the release system, I@PDPAM has a value of no public access, meaning that all datasets created are private. The values for the other possible options (execute, read, write, and maintenance) are also defined in USERI@P.

3.5.2.4 Preparing the user community

The next step towards privacy is to prepare the user community. Fundamentally, the only difference between a private and a nonprivate system is the treatment of the ownership value.

In a nonprivate system, ownership is recorded with a dataset when it is saved but is not used as a criterion when the dataset is accessed. If ownership information is supplied by the user on a nonprivate system, it is ignored without comment. Ownership can be recorded for existing datasets by means of the SETOWN utility (see next subsection).

All other privacy-related parameters (public access mode, access tracking, and permits) can be specified and are saved with the dataset. The privacy parameters do not have any effect, however, until privacy is turned on. Use the MODIFY control statement to set these values for already existing datasets.

3.5.2.5 SETOWN utility

Use the SETOWN utility on a nonprivate system to set the ownership value in permanent datasets. With it, users can claim their datasets before privacy is enabled. This utility needs to know what the ownership value is going to be, therefore the ownership value must be defined prior to bringing up the new COS. Again, setting the ownership value for a permanent dataset with SETOWN has no effect on the accessibility of the dataset until privacy is enabled.

The behavior of the privacy-related parameters and control statements on a nonprivate system allows a site to convert all COS Job Control Language (JCL), programs, and permanent datasets to privacy mode at leisure. It also facilitates testing by making it possible to switch back and forth between a nonprivate and a private system, or private and a dedicated time system without changing code, COS JCL, or the permanent dataset base.

NOTE

When switching back and forth between a nonprivate and a private system, care should be taken to avoid creating datasets that are identical in every respect except ownership because they might become inaccessible on a nonprivate system.

The final step to implementing permanent dataset privacy is to claim all unclaimed datasets by giving them a standard owner value using the SETOWN utility. Claiming unowned datasets in this manner helps operational personnel locate those inevitable datasets that disappear when privacy is enabled. Any datasets still lacking owner identification when the private system is deadstarted are claimed during startup using the default system owner value.

Once these procedures have been completed, the system is ready for full-time privacy.

3.5.3 PRIVACY AND THE PERMANENT DATASET UTILITIES

The three permanent dataset utilities PDSDUMP, PDSLOAD, and AUDIT support permanent dataset privacy. These utilities restrict average users to operations on their own datasets.

Restrictions are placed on the PDSDUMP, PDSLOAD, and AUDIT utilities when privacy is enabled. The OWN and NOWN parameters require the presence of the CW parameter. When privacy is not enabled, the CW parameter is not required with OWN and NOWN. This arrangement permits testing the privacy related features (AUDIT by owner, for example) on a nonprivate system.

NOTE

The JOB or ACCOUNT statement US value ceases to be an implied dataset selection criterion on a private system. If US is to be part of the dataset selection criteria, it must be supplied on the utility control statement.

The US parameter cannot be specified without also supplying CW on nonprivate systems.

A full AUDIT is obtained by supplying the CW parameter but not the US parameter on a nonprivate system and is no longer obtained by running the job under US=SYSTEM. That is, the discussions of the PDSDUMP, PDSLOAD, and AUDIT utilities apply to a nonprivate system if one substitutes US for OWN in them.

3.5.3.1 PDSDUMP utility

Use the OWN and CW parameters on the PDSDUMP control statement to determine catalog access. Table 3-1 summarizes the interaction of CW and OWN.

Table 3-1. CW and OWN Parameters on the PDSDUMP Statement

	OWN	No OWN
CW	If CW is specified, OWN becomes simply another search criteria (dumps OWN's catalog regardless of permission).	If OWN is not given, all datasets meeting the other control-card-supplied search criteria are dumped (a PDSDUMP with only CW supplied results in a dump of all datasets listed in the catalog).
No CW	Illegal	If neither OWN nor CW are specified, then the dump is restricted to the job owner's catalog (OWN defaults to the job owner value).

The selection of specific datasets is also affected by the other parameters of the PDSDUMP control statement.

3.5.3.2 PDSLOAD utility

The OWN and NOWN parameters are part of the PDSLOAD control statement. The OWN parameter identifies ownership value for datasets to be loaded, and a NOWN parameter allows selected datasets to be loaded to an owner different from the one who created it. If you specify either parameter, the CW parameter must also be supplied. The interaction between OWN, NOWN, and CW is similar to that between CW and OWN on the PDSDUMP control statement. Tables 3-2 and 3-3 summarize the interaction.

Table 3-2. PDSLOAD with CW

	No OWN	OWN
No NOWN	All datasets loaded to original owner's catalog	OWN's datasets loaded to OWN's catalog
NOWN	All datasets loaded to NOWN's catalog	OWN's datasets loaded to NOWN's catalog

Table 3-3. PDSLOAD without CW

	No OWN	OWN
No NOWN	Job owner's datasets loaded to job owner's catalog	Illegal
NOWN	Illegal	Illegal

The selection of specific datasets for loading is affected by the other parameters of the PDSLOAD control statement.

When the dataset being loaded has no ownership value in the load PDD found on the dump tape (normally the case on dump tapes created before privacy is installed), the dataset is loaded to the job owner's catalog unless NOWN has been supplied.

3.5.3.3 AUDIT utility

The AUDIT utility uses both the CW and OWN parameters and they have meanings that are similar to those used by PDSDUMP. Table 3-4 summarizes the interaction of these two parameters.

Table 3-4. AUDIT Parameters

	OWN	No OWN
CW	Own's catalog regardless of permission	Entire system catalog
No CW	Illegal	Job owner's catalog

The selection of specific datasets for auditing is affected by the other parameters supplied on the AUDIT control statement.

NOTE

The JOB or ACCOUNT statement US parameter has no significance to AUDIT on a private system. US must be specified on the AUDIT control statement if it is to be a part of the dataset selection criteria.

A full AUDIT is obtained by supplying CW alone with no specially cased owner values. Similarly, on a nonprivate system, US=SYSTEM no longer has the special significance it had in the past.

3.6 MASTER AND BACKUP CATALOG DATASETS

The Master and Backup Catalog features offer performance and convenience improvements to permanent dataset access. These improvements may offer a decrease in permanent dataset access times by a factor of 5 or 6 on moderately populated systems (10,000 dataset editions cataloged). Similar improvements can be found with the AUDIT control statement. The Master Catalog also allows you to use the nonlocal form of the DELETE control statement. The Master and Backup Catalog features are required if permanent dataset archiving is to be implemented; they are recommended for all sites.

The Master Catalog is essentially an index to the Dataset Catalog and to the Backup Catalog. It contains an entry for each dataset on the system. Each entry consists of two parts: a fixed-size section, which records dataset identification information, and a variable-length trailer section, which contains a varying number of structures called edition records. Each edition record contains pointers to the Dataset Catalog and/or Backup Catalog entries for one edition of the dataset. Thus, when the Master Catalog is present, the DSC is no longer searched, resulting in a substantial reduction in system overhead when a dataset edition is accessed. In an archiving environment, the Master Catalog also indicates whether the dataset edition is on-line or off-line.

The Backup Catalog contains information about the off-line copy of a dataset edition. In an archiving environment, the off-line copy may be the only copy of the dataset (as when the dataset edition is migrated or retired). If archiving is not implemented, the Backup Catalog is not actually used. However, some utilities, such as AUDIT, attempt to access the Backup Catalog and generate a warning message if the access is unsuccessful. For this reason, a token Backup Catalog should be created if the Master Catalog is present.

3.6.1 CATALOG CREATION AND RECOVERY UTILITIES

Seven utilities provide Master and Backup Catalog creation and recovery functions. The following is a list of these utilities, along with their description:

Utility	Description
GENBCD	Generates a new Backup Catalog
VALBCD	Validates and recovers an existing Backup Catalog
ALTBCD	Extends an existing Backup Catalog
GENMCD	Generates a new Master Catalog
STATMCD	Reports Master Catalog utilization statistics
PDMCAT	Transfers control of the Backup and Master Catalogs to the operating system
LOADCAT	Copies catalog datasets from backup media to on-line disk.

These utilities replace and extend the functions provided by the GENCAT utility released in the COS 1.15 and 1.16 systems. The standard build procedures released with the system generate the new utilities. For information on how to use these utilities, see the Operational Aids Reference Manual, publication SM-0044.

Most of these utility programs are intended for execution as part of a STARTUP class job, often referred to as the GENCAT job. They are an extension of the startup process, but they may be executed during normal production as long as the catalogs created or recovered are not transferred to the operating system.

The following sample jobs indicate the way the utilities are to be used together. Separate sections following the examples provide guidelines for catalog sizing.

Example 1:

Example 1 shows the trial sizing of the Master Catalog. This job can be run during ordinary batch production because it does not transfer the trial catalogs to the operating system.

```
JOB     (JN=TRIAL,T=64,MFL=512000)
ACCOUNT (AC=acn,APW=apw,US=usn,UPW=upw)
*
*       Step 1: Create a dummy Backup Catalog.
*
ACCESS  (DN=GENBCD)
GENBCD  (SIZ=1)
*
*       Step 2: Create a trial Master Catalog.
*
ACCESS  (DN=GENMCD)
GENMCD  (REG=reg,SIZ=siz)
*
*       Step 3: Get Master Catalog statistics.
*
ACCESS  (DN=STATMCD)
STATMCD .
*
*       Step 4: Delete the dummy catalogs.
*
DELETE  (DN=$BCD)
DELETE  (DN=$MCD)
```

This example creates a dummy Backup Catalog with a size of 1 block, satisfying the GENMCD requirement. It then executes GENMCD, specifying a trial number of regions and region size. GENMCD accesses the Dataset Catalog internally. Even though the DSC is active, the results will be close enough for the purpose. Next, it runs STATMCD to produce the region utilization report. Finally, it deletes the permanent datasets created by GENBCD and GENMCD because these catalogs will not be used again.

Also, the number of regions for GENMCD should be a prime number. The following is a list of prime numbers:

1	2	3	5	7	11	13	17
19	23	29	31	37	41	43	47
53	59	61	67	71	73	79	83
89	97	101	103	107	109	113	127
131	137	139	149	151	157	163	167
173	179	181	191	193	197	199	211
223	227	229	233	239	241	251	257
263	269	271	277	281	283	293	307
311	313	317	331	337	347	349	353
359	367	373	379	383	389	397	401
409	419	421	431	433	439	443	449
457	461	463	467	479	487	491	499
503	509	521	523				

Example 2:

Example 2 shows the initial creation of the Backup Catalog. This job can also be run during production batch because it only initializes the BCD.

```
JOB      (JN=TRIAL,T=64,MFL=512000)
ACCOUNT  (AC=acn,APW=apw,US=usn,UPW=upw)
*
*        Initialize the Backup Catalog.
*
ACCESS   (DN=GENBCD)
GENBCD   (SIZ=siz,ED=4095)
```

This is the minimum required to initialize the BCD. To determine the starting size (the *siz* parameter), see Sizing the Backup Catalog in this section. In addition to the recommended ED=4095, read, write, and maintenance control words can also be specified, as can PDN and ID. The default PDN=$BACKUP and null ID are created in this example.

Example 3:

Example 3 is the standard GENCAT job. This job recovers the Backup Catalog, creates the Master Catalog, and transfers control over them to the operating system. It must be run as a STARTUP class job.

```
JOB      (JN=TRIAL,T=64,MFL=mfl)
ACCOUNT (AC=acn,APW=apw,US=usn,UPW=upw)
*
*         Step 1: Recover the Backup Catalog.
*
ACCESS   (DN=VALBCD)
ACCESS   (DN=$BCD,PDN=$BACKUP,ED=4095,UQ)
VALBCD   .
*
*         Step 2: Delete the previous copy of the
*                 Master Catalog.
*
ACCESS   (DN=$MCD,PDN=$MASTER,ED=4095,UQ,NA)
DELETE   (DN=$MCD,NA)
RELEASE (DN=$MCD)
*
*         Step 3: Create a new Master Catalog.
*
ACCESS   (DN=GENMCD)
GENMCD   (REG=reg,SIZ=siz,ED=4095)
*
*         Step 4: Run the MCD statistics program.
*
ACCESS   (DN=STATMCD)
STATMCD .
*
*         Step 5: Transfer catalogs to system control.
*
ACCESS   (DN=PDMCAT)
PDMCAT   .
```

This job specifies the minimum control statement parameters and the suggested ED=4095 for both catalogs. Sites using the archiving feature usually add the appropriate accesses and submits for the backup, space manager, and recall jobs after the PDMCAT statement. None of the utilities are in the SDR, and DELETE,DN must be used until after the completion of the PDMCAT step. The default PDN=$BACKUP and null ID are created in this example.

Example 4:

Example 4 alters the size of the Backup Catalog. This job is the standard GENCAT job, as well as a call to ALTBCD. It could actually be used as the standard GENCAT if desired.

```
JOB      (JN=TRIAL,T=64,MFL=mfl)
ACCOUNT (AC=acn,APW=apw,US=usn,UPW=upw)
*
*         Step 1: Recover the Backup Catalog.
*
ACCESS   (DN=VALBCD)
ACCESS   (DN=$BCD,PDN=$BACKUP,ED=4095,UQ)
VALBCD   .
*
*         Step 2: Enlarge the Backup Catalog.
*
ACCESS   (DN=ALTBCD)
ALTBCD   (SIZ=newsiz)
*
*         Step 3: Delete the previous copy of the
*                 Master Catalog.
*
ACCESS   (DN=$MCD,PDN=$MASTER,ED=4095,UQ,NA)
DELETE   (DN=$MCD,NA)
RELEASE  (DN=$MCD)
*
*         Step 4: Create a new Master Catalog.
*
ACCESS   (DN=GENMCD)
GENMCD   (REG=reg,SIZ=siz,ED=4095)
*
*         Step 5: Run the MCD statistics program.
*
ACCESS   (DN=STATMCD)
STATMCD  .
*
*         Step 6: Transfer catalogs to system control.
*
ACCESS   (DN=PDMCAT)
PDMCAT   .
```

The output from each GENCAT job execution should be examined to determine when Backup Catalog (and Master Catalog) space is getting low. The job can then be modified to specify new sizes, and the startup parameter file can be changed to specify that this job is submitted instead of the normal GENCAT job at the next restart (unless this job is the normal GENCAT job). The percentage of size increase will be tailored to the frequency of restarts (the PM schedule), and the historical rate of catalog utilization.

This job can be used as the normal GENCAT job because ALTBCD does nothing if the specified catalog size is the same as the current size. In this case, submission of the archiving jobs, if any, follows the PDMCAT job step.

Example 5:

Example 5 shows how to load catalog datasets with the LOADCAT utility.

```
JOB     (JN=LOADCAT,T=8,*TAPE=1)
ACCOUNT (AC=acn,APW=apw,US=usn,UPW=upw)
*
*  Restore the Backup and Backup Volume Catalogs.
*
ACCESS  (DN=LOADCAT)
LOADCAT .
<EOF>
ONLINE,VOL=B4356,ORG=SQ,LB=SL,DT=*TAPE.
BCD,PDN=$BACKUP,ID=COSARCH,ED=4095.
BVCD,PDN=$BVCD,ID=COSARCH,ED=4095.
<EOF>
```

In example 5, the Backup Catalog and Backup Volume Catalog are restored from TAPE volume B4356; they become permanent datasets with edition number 4095. The generic resource name appears on both the JOB statement and the ONLINE directive; it should be changed to match the local configuration. This job can be run at any time because it does not invoke GENMCD or transfer the catalogs to system control.

3.6.2 SIZING THE BACKUP CATALOG

The Backup Catalog size is defined in terms of blocks, each block containing eight entries. Every backed-up dataset edition requires one entry. In addition, when a dataset edition is migrated or retired, any DXT entries for that edition are transferred to the Backup Catalog. Thus the minimum size required for the Backup Catalog is as follows:

((NUDSED + NUDXT + 1) / 8) * GROWTH

NUDSED — The number of user dataset editions registered in the Dataset Catalog.

NUDXT — The number of DXT entries in use.

GROWTH — A multiplier based on the rate of growth in dataset editions cataloged over an arbitrary time span. If this growth factor is unknown, start with a small integer, such as 2, and monitor the Backup Catalog utilization reported by VALBCD.

3.6.3 SIZING THE MASTER CATALOG

Determining the size of the Master Catalog can be complicated. The Master Catalog is divided into equal-sized units known as regions. Each region is divided into 32-word entries (16 entries per block). The first *n* entries in each region are used as a key table, with the remainder used for datasets. Unlike the DSC, which has an entry for each dataset edition, the Master Catalog contains only one main entry for a given PDN, ID, and OWNer. The main entry registers up to four editions of the dataset; continuation entries can record up to seven additional editions.

Every dataset owner has an assigned region into which all of the datasets are cataloged. This assigned region is known as the owner's home region and is determined by a hashing algorithm applied to the owner name. If the owner's home region should become completely filled, additional dataset editions will be cataloged in the next sequential region that is not full (the first region of the catalog is considered to follow the last). Thus, the Master Catalog is not considered full until all entries in it are in use.

Ideally, every dataset belonging to a given owner should be cataloged in the owner's home region. This minimizes the time and I/O required to find a given dataset edition. In practice, some owners have large numbers of datasets and/or editions, while others have few. Thus, a Master Catalog meeting the ideal may be quite large, with a low average utilization. The purpose of sizing the Master Catalog is to find some acceptable compromise between catalog capacity and search time, or to minimize the number of region overflows.

Because each site's permanent dataset profile is different, no universal formula can be applied to the Master Catalog sizing problem. Trial and error testing, using something like the job illustrated in example 1 above, is the most practical approach. An initial guess can be derived from the following:

REG = *nown* / 4
SIZ = (((*maxds* + (*maxed* / 7)) / 15) * 4

where

nown The number of owners cataloged (in the default system, the number of user numbers registered in the $VALIDATION dataset)

maxds The maximum number of datasets (PDN,ID) cataloged by any owner on the system

maxed The maximum number of editions of any dataset on the system (except perhaps $SYSTEMLOG)

This formula overestimates the required size of the Master Catalog. You can then reduce the number of regions and region size using the job in example 1. Aim for a catalog size that produces no more than 10 to 25 percent of regions overflowed; the fewer the better. This range allows for some growth in dataset editions cataloged without producing serious performance penalties. Monitor the STATMCD report while it is in production to determine when the Master Catalog should be enlarged.

3.6.4 THE STARTUP JOB CLASS

The released system contains a STARTUP job class as part of the default job class structure. Nearly all sites define their own job class structures by using the JCSDEF utility. Sites using the catalog creation utilities must supply a STARTUP job class definition to JCSDEF. The ZSUB characteristic enables the unique environment required by GENMCD to guarantee that the Master Catalog reflects the state of all user dataset editions and system spooled datasets cataloged in the DSC and BCD.

3.7 PERMANENT DATASET ARCHIVING

Permanent Dataset Archiving provides an automated backup facility, integrated with a permanent dataset space management facility. Five major programs implement these facilities in conjunction with function-level support from the Permanent Dataset Manager (PDM) system task. See the Operational Aids Reference Manual, publication SM-0044, for more information. The following subsections provide an overview of these programs.

3.7.1 DATASET BACKUP

BACKUP creates copies of all on-line permanent datasets that have been created or modified since the previous backup. Modifications include not only changes to the dataset content but also any changes to the catalog information for a dataset.

Back-up volumes are written in the same format used for the PDSDUMP and PDSLOAD utilities. Thus, any back-up volume generated could be reloaded by PDSLOAD, if desired.

The back-up processor also generates copies of the Master Catalog Dataset (MCD), the Backup Catalog Dataset (BCD), and the Dataset Catalog/Dataset Catalog Extension (DSC/DXT). Use the back-up copies of these catalogs as input to several utilities.

3.7.2 SPACE MANAGEMENT

Space management attempts to guarantee that a certain minimum amount of on-line disk space is available for allocation. This goal is achieved by dataset deletion, retirement, and migration.

Deletion consists of releasing the dataset edition's on-line allocation and removing its entry from the system catalogs.

Both retirement and migration consist of releasing the on-line allocation of a dataset edition and removing its entry from the on-line Dataset Catalog (DSC) but not from the Master and Backup Catalogs. As a result, the dataset edition remains available on its back-up volume.

The distinction between retirement and migration concerns availability. A retired dataset edition must be explicitly requested before access; a migrated dataset is implicitly requested when accessed. The migration and recall process is almost invisible to the user; the only clues are the AUDIT report and a delay message when the dataset is accessed.

All on-line datasets with current back-up copies are considered potential candidates for retirement or migration unless they have been excluded by either preferred residency or site-supplied directives to the space manager. Site-supplied directives determine dataset selection from the candidate list.

3.7.3 DATASET RECALL

RECALL and the RECALL control statement locate the off-line version of a permanent dataset and restore it to on-line direct access storage devices. A request to access a migrated dataset causes automatic activation of the recall job; the job doing the access is placed in suspended state until the dataset has been recalled.

RECALL also is involved in the restoration of retired datasets. It processes retired datasets when there are no requests outstanding for the recall of migrated datasets.

3.7.4 BACK-UP VOLUME CLEANUP

When sequential devices, such as magnetic tapes, are used as the back-up media, the normal system activity of deleting datasets and creating new ones tends to create a large number of back up volumes with little active data on them. Back-up volume cleanup is intended to remedy this situation by collecting active data from sparsely populated volumes and creating new densely populated volumes from that data. Volumes with no active data can then be recycled. This compaction occurs, however, only when enough data can be collected from volumes with an occupancy below a site-defined threshold to make up a new volume that has an occupancy above another site-defined threshold.

Cleanup can also perform a number of other functions. It can delete datasets from the back up volumes based on site-supplied criteria such as age. When back up makes duplicate volumes, cleanup can make new duplicates if one or more of the original set is corrupted. It also removes dataset catalog back up entries if all of the back up volumes for that dataset have been deleted from the Backup Volume Catalog.

3.7.5 DEVICE RELOAD

The RELOAD utility allows privileged users, such as the operator, to initiate a reload of datasets that were on a particular device or devices at the time of the most recent catalog dump. To be reloaded, the datasets must have been on-line. Retired and migrated datasets remain retired or migrated.

Directives to the utility allow the site to specify whether all or a subset of the previously on-line dataset base is to be reloaded. Any datasets that were previously on-line but have been excluded from the reload process by directive become migrated.

3.7.6 INSTALLATION

All of the archiving feature utilities assume that both privacy and security are enabled. They all require the SCRDSC (Read Dataset Catalog) and SCPDAD (Permanent Dataset Administrator) privileges for execution. In addition, RECALL/RECIO requires SCQDXT (Link DXT) privileges under certain circumstances. For maximum security, it is strongly recommended that archiving software be run under a special user number that has these privileges rather than assigning the privileges to the executable binaries themselves. The standard generation procedures assume this convention.

The source code for all archiving utilities resides in UTILPL, as does the source for the library $ADMLIB. Each utility is contained in a deck that has the same name as the utility. $ADMLIB is made up of a number of separate decks beginning with the dummy deck ZZZADMON and ending with the dummy deck ZZZADMOF. UTILPL also contains a number of common decks, all beginning with the letters ADM, which are called by both the utilities and $ADMLIB.

Most of the catalog generation and recovery utilities use the $PDSLIB library instead of $ADMLIB for catalog I/O. The source for this library resides in the PDSLBPL update program library and contains both subprograms and common decks. All products that call $PDSLIB routines use PDSLBPL as a common library during UPDATE to acquire the common decks.

3.7.7 SITE CONFIGURATION

While most of the archiving feature is self-configuring, based on directive inputs to the various utilities, a few site specific parameters must be supplied in a local configuration mod applied to UTILPL before the utilities are generated. All of these parameters concern the I/O helper jobs submitted internally by the BACKUP, RECALL, and CLEANUP utilities.

1. The maximum number of I/O helper jobs that any of the three utilities may submit at one time is controlled by the Fortran parameter TAPEJOBS found in UTILPL common deck ADMCNFIG. In the released system, the value of TAPEJOBS is 2. You can change this with the following mod in which,

   ```
   *DELETE ADMCNFIG.9
      PARAMETER (TAPEJOBS = n)
   ```

   n The maximum number of I/O helpers. This number should be less than or equal to the number of on-line tape drives.

2. The job and account control statements for the I/O helper jobs are generated by $ADMLIB subroutine SPINTAPE. This routine uses the account and user numbers from the Job Account Table. Because SPINTAPE cannot determine the user password, it examines the value of a 2-word array declared in UTILPL common deck ADMCNFIG. In the released system, the value in this array is 0s, and SPINTAPE assumes that the user password is the same as the user number. You can redefine the password with the following mod, in which

   ```
   *DELETE ADMCNFIG.18
      DATA OPSUPW / 'upwl', 'upw2' /
   ```

   'upwl' The first 8 characters of the user password and *'upw2'* is the last 7 characters of the user password, blank filled as necessary.

SPINTAPE does not generate an account password field. If necessary, SPINTAPE itself must be modified. SPINTAPE is written in Fortran.

3.7.8 FRONT-END TAPE SERVICING

If the BACKUP, RECALL, or CLEANUP jobs are submitted from a STARTUP class job or from the operator's console, and front-end tape servicing is enabled, one of several options must be used to ensure that the correct front end is involved in helper I/O job tape mount requests. Front-end servicing is usually automatic if the jobs are submitted from the servicing front end.

If the servicing front end does not require station slot, the simplest option available is to specify the source ID of the appropriate station on the submit control statement. Alternatively, one of the TQM options, such as the default servicing front-end definition, can be used. This latter option has the advantage of working with console submitted jobs as well as with those submitted from JCL.

If the servicing front end requires station slot, two other options must be used. The simplest is to submit the BACKUP, RECALL, and CLEANUP jobs from the front end to provide the servicing. Alternatively, the Station Slot Authorization File feature can be used to predefine the appropriate station slot for the user number under which the archiving jobs will be run. The latter option must be combined with one of the techniques described in the previous paragraph for ensuring the correct servicing front-end system.

3.7.9 PREPARING THE SITE FOR ARCHIVING

This section provides some general guidelines for operating with the archiving facility. Because operating conditions vary so much from site to site, they should not be regarded as hard and fast rules; in fact, every site that has installed this software uses it in unique ways, some making significant modifications to the code. The following is a list of guidelines to consider.

- Write a mod for any site configuration changes relating to archive job validation and accounting, the number of I/O helper jobs, and front end tape servicing parameters. Then build or rebuild any components affected by the mod.

- Create a user number under which all of the archiving jobs will be run. This user number requires the SCPDAD, SCRDSC, and SCQDXT privileges at minimum; SCRESON may also be needed.

- Initialize the Backup Volume Catalog. This involves running the BVCEDIT utility. Note that all of the volumes must be identical in terms of label type, character set, recording density, and length. Undersize the volumes to allow for surface defects that cause erase gaps. For a 2400 foot, 6250 b/i tape, use SIZE=20000000, and for a cartridge tape, use SIZE=24000000. These numbers are suggested for high-quality tapes; use less for tapes of uncertain surface. It is not necessary to prelabel the tapes if they are blank because the archiving I/O jobs initialize the labels whenever the tapes are used as output volumes.

- Write the GENCAT job, using the examples given earlier as a starting point. It is very convenient to have the GENCAT job submit the RECALL job, as well as any other cyclic archiving jobs.

- Put copies of key archiving components on the IOS expander disk or expander tape, along with a job that can be submitted from the operator's console to move the components back to the Cray system in the event of a forced-install or device-down situation. Key components include the catalog generation and recovery utilities LOADCAT, VALBCD ALTBCD, GENMCD, STATMCD, and PDMCAT, as well as the RECALL, RECIO, PDSLOAD, and RELOAD utilities, and all of the jobs used to execute these utilities. Consider using TEDI to modify the jobs and/or their directives.

  The job stored on the IOS to restore these key components should be structured in such a way that it can bypass datasets that are already on-line (rather than creating a new edition). The easiest way to do this is to acquire a specific edition of each utility or job dataset. Edition 4095 is recommended for the current version of these datasets because it requires a deliberate operation when newer versions are substituted. However, any edition number will do as long as all personnel involved in operations know what the current edition is.

- Add to the System Directory the job ACCESS and ENTERs for the I/O helper programs BUPIO, RECIO, and CLUPIO, as well as for the user-oriented utilities RETIRE and RESTORE. It is strongly recommended that, for security reasons, the main archiving programs and their helpers get their privileges from the user executing them rather than from the LDR or SEGLDR GRANT parameters. The exceptions are RETIRE and RESTORE; because these utilities can be called by the users, their privileges must come from LDR/SEGLDR GRANT parameters.

- Develop a mechanism for recovery of the SDR components. This usually takes the form of a PDSDUMP tape of all current SDR modules. The tape is used to expedite recovery from a device-down or forced-install situation.

- Determine the directive inputs needed to accomplish site goals for the package. Most of these decisions concern issues such as manual versus automatic operation, activation schedules, and the identification of the Backup Volume Catalog; the directives are described in the Operational Aids Reference Manual, publication SM-0044. You must also specify OMIT directives to the BACKUP and MANAGE programs for certain specific datasets.

  $RD (Resource Dataset) $VALIDATION, $ACCOUNT, and $BULLETIN must be omitted from backup processing to avoid a deadlock between BACKUP and its BUPIO helper jobs. Also omit the highest edition of the $SYSTEMLOG dataset so that system log datasets will be backed up only when they are completely full. If, as recommended, a separate backup procedure exists for the SDR components, they may also be omitted.

  Some separate procedure must be established for the backup of $RD $ACCOUNT, $VALIDATION, and $BULLETIN. Typically this takes the form of a PDSDUMP job, which could be combined with the SDR component dump if desired. The backup media could be on-line tape, a front end system, or the IOS expander devices. A job should be created for PDSLOADing these datasets and saved on the IOS expander disk.

  Some datasets should never be migrated or retired by MANAGE. These include the archiving utilities that are not in the $RD SDR, $VALIDATION, $ACCOUNT, and $BULLETIN. The highest edition of the CRAY1SYSTEMDUMP dataset should also be kept on-line to avoid creating duplicate editions (STARTUP saves this dataset before GENMCD runs, so PDM does not know about any of the off-line editions). There are two ways to make datasets immune to migration or retirement. The first is to SAVE or MODIFY them, specifying RESIDE=ONLINE; to do this, the user must have the SCRESON privilege. The second method is to use OMIT directives to the MANAGE program. This method is especially convenient because the omit list is essentially the same as that for the BACKUP program. It should be noted here that SDR datasets and the datasets accessed by the system, such as $SDR, SYSROLLINDEX, and JOBCLASSROLLED, and all of the system catalogs, are protected from space management by PDM.

- Plan for the first run of the BACKUP program, because the first run is effectively a full system dump. Note, however, that BACKUP cannot dump any datasets that are uniquely accessed at the time it is executing. It may be desirable to do the first BACKUP with as few user jobs in the system as possible.

Subsequent BACKUP runs are incremental, so the volume of data is much smaller. The scheduling of normal BACKUP runs is usually determined by the rate of dataset creation and/or modification at the site. More frequent runs provide greater database integrity by keeping the backup media current; they also provide the MANAGE program with a greater number of potential candidates because datasets cannot be space managed until they have current backup copies. On the negative side, frequent BACKUP runs may produce a lot of sparsely populated tape volumes, with the resulting need to run the CLEANUP program more often. In addition, BACKUP(BUPIO) interlocks all of the datasets being written to a tape volume for the length of time it takes to write the volume. User jobs trying to access an interlocked dataset are placed in wait event until the interlock is cleared. The duration of the interlock can be anywhere from a few minutes to several hours, depending on the availability of the hardware and/or operators. For these reasons, may sites run BACKUP only once a day, usually on the third shift, when the job load is lighter.

- When determining which thresholds to define for MANAGE, take into account the fact that MANAGE cannot release disk space instantly. Thus, the upper threshold has to be low enough that MANAGE has a chance to execute before the system runs out of disk space. An upper threshold above 95% is not recommended.

- When planning SELECTs and OMITs for MANAGE, consider establishing a rule regarding the number of editions of a given dataset allowed to exist, and how many may be retained on-line. Many sites use MANAGE to enforce such a rule by deleting all but the '*n*' highest editions and migrating all but the highest edition. This technique often yields enough disk space to make weighted migration unnecessary. Weighted migration should still be configured, however.

  The measure of space management success is the number of recalls that result from migration. The fewer recalls the better. Weighted migration provides a mechanism for selecting datasets based on statistical data that has some predictive value about the likelihood that a dataset edition will be accessed in the future. This is especially true for AGE (since last access) and ACCESS (count); the third variable, SIZE, has no predictive value, but it has some use as a tie breaker in the event that two candidate dataset editions are otherwise identical. Some experimentation is necessary at each site. For starting values, try AGE=100, ACCESS=50, and SIZE=25; these values bias the selection algorithm heavily in favor of the time since last access, with a moderate sensitivity to the frequency of access, and very little consideration as to the size of the dataset.

- If at all possible, relegate CLEANUP runs to light usage hours. CLEANUP effectively locks out BACKUP and RECALL while it is running. Some sites actually drop or suspend BACKUP and RECALL before starting CLEANUP, especially if there is a good chance of contention for tape drives or other hardware. CLEANUP requires a significant amount of wall-clock time because it must usually read many tapes to produce one output volume. Depending on the contents of the volumes being compressed, CLEANUP may make two passes over the same tapes. As an aid to the operators, CLEANUP automatically writes a report to the IOS printer, giving a list of the volumes it will ask for in request order.

- Make sure that the operators understand the basic scheme of the archiving software, the names of the various jobs, and above all that archiving jobs BACKUP, MANAGE, RECALL, CLEANUP, and their I/O helpers must never be killed (use DROP instead because KILL is permanent).

- Establish some procedure for preserving the volume names to which BACKUP and CLEANUP dump the system catalogs. It is essential to know which volume(s) contains the most recent dump of the system catalogs should an install or down-device restart be necessary. Use the REPORT directive to BACKUP and CLEANUP to route execution time output to either the IOS printer or some front-end system. Either preserve these reports or have the volume name(s) entered in the console logbook.

3.7.10 PERFORMING AN INSTALL WITH ARCHIVING

An install is necessary any time the master device has been lost, or whenever you need to increase the size of the Dataset Catalog or system dump area. The archiving software has built-in facilities for recovering the permanent dataset base after an install. The following is a suggested procedure; local modifications are almost always necessary. These suggestions assume that all of the tasks covered in Preparing the Site for Archiving, in this section, have been completed.

- Perform the install startup. The only modifications needed to the normal restart parameter file are the replacement of *RESTART with *INSTALL, and the removal of any *SUBMIT directives. If the last edition number of the $SYSTEMLOG dataset is known, force the edition number (*SYSLOG directive) to that edition (this assumes that the highest edition of the system log dataset is omitted from BACKUP); this will keep the $SYSTEMLOG edition numbers sequential and without gaps.

- When the machine is up, submit from the operator's console the IOS job that acquires the key archiving components and their executing jobs from the expander disk or expander tape.

- Load the $ACCOUNT, $VALIDATION, $RD, and $BULLETIN datasets from their back-up media. It is essential that this be done before the execution of GENMCD, because GENMCD marks the datasets as being migrated, and any subsequent jobs (including RECALL itself) go into wait event at job initiation time.

- Determine which backup volume has the most recent catalog dumps on it. Then edit the LOADCAT job to restore the $BCD and $BVCD catalogs as permanent datasets. Run the job.

- If the site runs special jobs for dumping such datasets as those in the SDR, perform their PDSLOAD at this time.

- Restart the system, using the normal restart parameter file. GENMCD now generates the Master Catalog, and all datasets that were on-line at the time of the last BACKUP or CLEANUP run are now marked as both migrated and reloadable.

- If the SDR datasets are not yet on-line, use RELOAD or RESTORE to initiate recall of the datasets. RELOAD is preferred because it can process datasets of any ownership, but RESTORE does the same if all SDR datasets have the same ownership.

- When the SDR datasets are back on-line, run the SDR generation job. Production can now resume, although nearly any dataset accessed by a job will result in a wait for dataset recall. If this is acceptable, the install is complete.

- Production is more normal if at least a part, if not all, of the datasets that were on-line at the time of the install are restored. The RELOAD utility is designed specifically for this purpose. By default, it causes the recall of all datasets marked as reloadable by GENMCD. Directives to reload also allow the site to specify that only a subset of reloadable datasets be recalled, such as those accessed within the last *n* days, or accessed more than a certain number of times. Datasets effectively omitted by these directives remain in migrated status, so they will be recalled automatically should a user access them in the future.

3.7.11 RECOVERING FROM A DOWNED DEVICE WITH ARCHIVING

Recovering from a downed device with the aid of the archiving software is essentially the same as the install procedure described previously, but with the added complication that one does not generally know exactly which datasets must be recovered. Fortunately, the status of only a few specific datasets must be checked manually.

- Perform a normal system restart. Edit the start-up parameter file to make the downed device NAVAIL and to remove the *SUBMIT directive for the GENCAT job.

- When STARTUP asks what to do with permanent datasets that reside on the downed device, tell it to delete them all.

- Submit the IOS job that acquires the key system components from the IOS. This ensures that any utilities or jobs that were previously on the down device are are now on-line.

- Using an AUDIT job or the console DAT command, determine whether the Backup Catalog ($BCD) and the Backup Volume Catalog ($BVCD) are on-line. If either or both are missing, restore them with the LOADCAT utility.

- Restore the $ACCOUNT, $VALIDATION, $RD, and $BULLETIN datasets if any of them have been lost. If the site has a separate dump of the SDR components, reload them.

- Restart the system, this time allow the GENCAT job to execute. All datasets that were previously on the downed device now appear as migrated. GENMCD will also have marked them as reloadable.

- If the SDR datasets have not been restored, run a RELOAD or RESTORE job to bring them back on-line. Then run the JSYSDIR job. Note that it is possible to submit the JSYSDIR job without doing the RELOAD or RESTORE first, but the job will recall the SDR datasets one at a time which is a potentially time consuming process.

- Production may now resume. However, any access of a dataset that was on the downed device results in an event wait while RECALL restores it. If a large number of datasets were lost, it might be desirable to run a RELOAD job to recall all or some of those datasets.

3.7.12 USER EXITS

User exits allow you to install local system code at key entry points in the operating system. These key points remain compatible between system levels.

The key entry points and the names of the local routines executed are the following:

Local Routine	Entry Point
$EXPNEP	Entry to EXP before processing user request
$EXPADV	Job-step advance
$EXPTRM	Job termination
$EXPINIT	Job initiation (EXP)
$EXPABT	Job step abort
$$EXPAQR	ACQUIRE processing
$EXPDSP	DISPOSE processing
$EXPFCH	FETCH processing
$JSHFL	Memory allocation/deallocation
$JSHINIT	Job initiation, JSH processing
$JCMASG	Job class management
$STGSAV	Stager dataset SAVE request (applies to input jobs and user datasets)
$STGDEL	Stager dataset DELETE request

The system is released with the user exits turned off. To turn on a user exit, remove the SKIP statement preceding a user exit definition in common deck COMUXSYM in the COSPL. You must also supply a routine with the expected symbol name. To find out more about the user exit macros, refer to the common decks UEXIT and COMUXSYM in COSPL.

The following is an example of a local user exit:

```
*ID UEXITAW,DC=.
*DK UEXIT1
         IDENT     TEST
*CALL COMSEG
         EXT       ERROR1
         MACRO
NAME     GENX
         LOCAL     MSG,ID
         ENTRY     NAME
NAME     $SUB      AREG=1:5
         S1        CTID,0
         EXT       BTAD
         R         BTAD
         MSG+ID,0  S1
         MSGQ      ADR=MSG
         J         $RETURN
MSG      BUILD     SM,LE,(TYPE=SMTYPIN,WC=SIZE)
         DATA      'NAME called by task'
ID       =         W.*-MSG-LE@SM
         BSSZ      1
SIZE     =         W.*-MSG-LE@SM
GENX     ENDM
$EXPNEP  GENX
$EXPADV  GENX
$EXPTRM  GENX
```

```
$EXPINIT  GENX
$EXPABT   GENX
$EXPAQR   GENX
$EXPDSP   GENX
$EXPFCH   GENX
$JSHFL    GENX
$JSHINIT  GENX
$JCMASG   GENX
$STGSAV   GENX
$STGDEL   GENX
          END
*MOVEDK UEXIT1:STPTAB
```

3.7.13 RESOURCE DATASET RECOVERY

The Resource Dataset information must be recovered after a deadstart or restart. The RDVAL utility performs this operation and should be run after the GENCAT job. RDVAL reads the master catalog created by the GENMCD utility to validate the permanent and archived dataset counts for all owners. The RDQSC utility generates the queue information that RDVAL needs to in order to validate the job counts for the user entries. The RDQSC utility requires an idle system to create the correct information. The RDVAL utility can also be run on a system without the archiving system being used. RDVAL can get the permanent dataset information from a binary audit from the AUDIT utility. This method is much slower than using the Master Catalog, therefore it is recommended that the GENMCD utility be used to generate the Master Catalog even when not running the archiving system.

Example 1:

Example 1 is a job that validates the resource dataset when submitted as a STARTUP job or run on an idle system:

```
JOB     (JN=TRIAL,T=64,MFT=mfl)
ACCOUNT (AC=acn,APW=apw,US=usn,UPS=upw)
*.
*.     Step 1: Run the RDQSC program to generate
*.             the queue information.
*.
RDQSC,B,MF.
*.
*.     Step 2: Run the RDVAL program to validate
*.             the resource dataset.
*.
RDVAL,UPDATE,LO=SHORT.
```

Example 2:

Example 2 is a job that validates the resource dataset included with a GENCAT job, submitted as a STARTUP job.

```
JOB     (JN=TRIAL,T=128,MFT=mfl)
ACCOUNT (AC=acn,APW=apw,US=usn,UPS=upw)
*.
*.     Step 1-n: Perform the GENCAT operations.
*.
*.
*.
*.     Step n+1: Run the RDQSC program to generate
*.               the queue information.
*.
RDQSC,B,MF.
*.
*.     Step n+2: Run the RDVAL program to validate
*.               the resource dataset.
*.
RDVAL,UPDATE,LO=SHORT.
```

Example 3:

Example 3 is a job that validates the resource dataset from a binary audit from the AUDIT utility, when submitted as a STARTUP job or run on an idle system.

```
JOB     (JN=TRIAL,T=64,MFT=mfl)
ACCOUNT (AC=acn,APW=apw,US=usn,UPS=upw)
*.
*.     Step 1: Run the AUDIT program to generate a binary
*.             audit of the permanent datasets on $BINAUD.
*.
AUDIT,CW=cpw,B.
*.
*.     Step 2: Run the RDQSC program to generate the
*.             queue information.
*.
RDQSC,B,MF.
*.
*.     Step 3: Run the RDVAL program to validate the
*.             resource dataset from the $BINAUD dataset.
*.
RDVAL,BA,UPDATE,LO=SHORT.
```

3.7.14 RESOURCE DATASET BACKUP

The resource dataset should be backed up on a regular basis. When the system is running with user limits active, the RDM task owns the resource dataset and the RDACC utility must be used to make a copy of the resource dataset. The copy can then be backed up. If the RDM task is not active, the resource dataset can be copied and backed up or backed up directly.

Example 1:

Example 1 is a job that generates a copy of the resource dataset that can be backed up regardless of whether the RDM task is active or not.

```
JOB     (JN=TRIAL,T=64,MFT=mfl)
ACCOUNT (AC=acn,APW=apw,US=usn,UPS=upw)
*.
*.    Step 1: Attempt to access the resource dataset.
*.
ACCESS,DN=$RD,PDN=RESOURCEDATASET,ID=SYSTEM,OWN=SYSTEM,UQ,NA.
*.
*.       The access succeeded - copy the dataset and save
*.       the copy.
*.
COPYD,I=$RD,O=RDC.
SAVE,DN=RDC,PDN=RESOURCEDATC1,ID=SYSTEM,PAM=pam,R=rp,M=mp,W=wp.
*.
*.       Exit.
*.
EXIT.
*.
*.       The access did not succeeded - create a copy using RDACC
*.       and save the copy.
*.
RDACC,NRD=RDC.
SAVE,DN=RDC,PDN=RESOURCEDATC1,ID=SYSTEM,PAM=pam,R=rp,M=mp,W=wp.
*.
*.       Exit.
*.
```

4. JOB SCHEDULER (JSH)

The information about job scheduler (JSH) internals covered in this section is intended to give the site analyst the ability to make decisions about tuning. For a complete discussion of JSH internals, see the COS Internal Reference Manual, Volume II: STP, publication SM-0141.

JSH tuning is important because each site has different tuning requirements. Some sites depend on a class structure to control job priority while others allow users to specify their own priority. Some sites run a large number of small jobs while others run a small number of large jobs. Some sites use a wide range of priorities while others run all jobs at the same priority. All of these factors have a bearing on scheduler tuning.

This section presents the material in the following order:

- Scheduling internals
  - CPU scheduling internals
  - Memory scheduling internals
- Installation parameters
- Tuning the JSH
  - Adjusting CPU scheduling
  - Adjusting memory scheduling
- Observing JSH performance

4.1 SCHEDULING INTERNALS

Of the several processes JSH executes, the two most important are memory scheduling and CPU scheduling. A user job contends for space in memory and its tasks contend for CPU time. Understanding how each process works and how the two processes interact leads to rational decisions about values for scheduler tuning parameters. CPU scheduling, the simpler of the two processes, is covered first.

4.1.1 CPU SCHEDULING

CPU scheduling is the process of allocating the CPUs among the user tasks eligible for execution. Any task associated with a job in memory that is not suspended is eligible to execute. These user tasks are kept by the JSH in a queue called the CPU queue.

Each user task in the CPU queue is given a time slice as it enters the queue. A time slice is the minimum unit of time the user task has to use the CPU (execute). Once connected to the CPU, a user task runs until its time slice is consumed or until a scheduling decision is made to switch the CPU to another user task.

Scheduling decisions are made with the following objectives:

- User tasks with high priority are given more access to the CPU than user tasks with low priority.
- I/O-bound user tasks are given the opportunity to use the CPU whenever they need it.

4.1.1.1 Meeting scheduling objectives

The way CPU scheduling objectives are met is fairly simple. Whenever JSH discovers that no user task is connected to the CPU, it searches the CPU queue, starting at the top, and connects the first user task not suspended for I/O to the CPU. When a user task exhausts its time slice, a new time slice is computed and the user task is moved to the bottom of the CPU queue.

The scheduler provides special treatment for user tasks suspended while waiting for I/O to complete. When a user task issues an I/O-suspend request, the user task does not relinquish its eligibility to be connected. Its position in the queue is not changed. All other suspend states cause the user task to lose its CPU queue residency. In practice, the CPU queue is effective. User tasks doing a lot of I/O tend to drift toward the top of the queue as they stay eligible for execution longer than other user tasks. User tasks with long time slices also tend to drift toward the top of the queue as user tasks with shorter time slices are disconnected and reentered at the bottom.

4.1.1.2 Time slice size

The size of a user task's time slice is directly related to its priority. The scheduler computes time slices using the following formula:

$$JXTS = max(I@TSMIN,(I@JST0 + I@JST1 * (P) + I@JST2 * (P^2) + I@JST3 * (P^3)) * (1 - I@SEMPEN * TCTWTS/TCTSTS))$$

where P is the job card priority, I@CJST0,..., I@JST3, I@TSMIN and I@SEMPEN are system tuning constants. TCTWTS is the time spent waiting on a semaphore in the last time slice; TCTSTS is the total number of cycles in the last time slice.

The site analyst can control the size of the time slice of a certain priority given to a user task by adjusting the coefficients in the time slice formula.

4.1.1.3 Scheduling exceptions

Two other elements of CPU scheduling are important and are exceptions to the general scheme described previously. The first exception is called the disconnect cost. Whenever a user task is disconnected from the CPU, such as for an I/O suspend, a small amount of time is subtracted from the user task's time slice. The reason for the subtraction is to prevent I/O-bound user tasks from remaining near the top of the CPU queue for long periods without using very much CPU time. The site analyst can adjust the size of the disconnect cost (see subsection 4.2, Installation Parameters).

The other exception is the way a user task is treated when it is placed in memory. The JSH ensures that each user task in memory gets an opportunity to use the CPU at least once before it loses its privilege to stay in memory. When a user task enters the CPU queue for the first time after gaining a memory allocation, the JSH places the user task at the top of the CPU queue and gives the user task an initial time slice. The initial time slice is the same for all user tasks and can be adjusted by the site analyst (see subsection 4.2.9, Initial Time Slice (I@JSITS)).

4.1.2 MEMORY SCHEDULING

When a job is not in memory and is not suspended, it is contending for memory. All jobs that contend for memory are kept by the scheduler in a queue called the memory request queue (MRQ). The MRQ is a priority ordered queue. A job in the MRQ has a memory priority identical to its job statement priority.

A job is removed from the MRQ when it is placed in memory. Like jobs in the MRQ, jobs in memory have a memory priority. A job is entitled to stay in memory until two conditions are satisfied:

- A job of higher memory priority is waiting at the top of the MRQ
- The job has been in memory at least as long as the in-memory thrash lock (defined next)

4.1.2.1 In-memory and out-of memory thrash locks

Thrashing is the term used to describe rapid, uneconomical movement of data between mass storage (disk) and memory. A thrash lock controls the amount of thrashing that can occur. The thrash lock is a means of keeping a job in (I@JSLK1 and I@JSLK4) or out of (I@JSLK2 and I@JSLK3) memory for a set amount of time to minimize the number of rolls experienced by the job and by the system.

A job in memory is guaranteed to stay in memory until the in-memory thrash lock expires. A job that is rolled out is guaranteed to stay rolled for the duration of the time specified by the out-of-memory thrash lock. The site analyst can adjust the size of the thrash lock value (see subsection 4.2, Installation Parameters).

4.1.2.2 Memory priority

As a job ages in memory, its memory priority decreases. That is, the longer a job stays in memory, the smaller its memory priority becomes. The memory priority of a job in memory is updated each time one of its tasks is disconnected. Disconnection occurs at the end of each time slice and whenever the user task is suspended.

The memory priority of a job in memory is determined with a formula illustrated in the discussion of memory priorities later in this section. Three aspects of memory priority can be adjusted:

- The lowest value that memory priority can reach
- The rate at which memory priority ages (called slope)
- The initial boost in memory priority that a job receives when the job is rolled in. The initial priority boost can be proportional to the size of the job.

When a job's memory priority ages to a value that is less than the priority of the job at the top of the MRQ, the scheduler checks to see if an allocation is possible. An allocation is possible only if sufficient memory is free or if sufficient memory is eligible to roll out to make room for the waiting job.

4.1.2.3 Expanding job size

An exception to the memory priority scheme is the behavior of the JSH when a job in memory attempts to expand. If sufficient memory is unavailable for expansion in place or by relocating the job to a larger free memory segment, the job is rolled out regardless of its memory priority or thrash lock.

The JSH allocates memory with consideration for expansion space. For an allocation to be possible, a specified amount of space must be left free after the allocation occurs unless fewer than a specified number of jobs are in memory. Expansion space pertains to all of memory, not to a single job. The space need not be contiguous with the current allocation. The site analyst can adjust the amount of expansion space and the number of jobs that must be in memory before expansion space is a necessary component of the allocation (see subsection 4.2, Installation Parameters).

4.1.3 DEMAND PROCESSING

Demand Processing is the process of mapping job priorities into discrete groups. These groups allow the job scheduler to impose absolute boundaries between jobs of different priorities.

The site analyst has the ability to give a particular job priority exclusive use of any of 3 machine resources: the CPU, memory, or I/O. Machine resources are not shared between groups unless there are resources that are not allocated. Priorities can also be grouped together into a single band. See section 4.2.17, Job Priority Band Table, for more information.

JSH allocates resources in a systematic way. The job(s) in the machine with the highest band value will get the resource before any job with a lower band value. All jobs that have the same band value for a particular resource share that resource according to the scheduling algorithms. JSH tries to assign unallocated resources to jobs in lower bands. Jobs with lower band values will not be able to hold resources if there are jobs with higher band values waiting for that particular resource. In other words, if a job with a higher band value is waiting for memory, jobs in memory that have a lower band value will be rolled out immediately, whether or not the job's in-memory thrash lock has expired.

Band values are fixed, unlike a job's memory priority which ages. Band values are set at job/task creation or when an operator changes the priority of a job.

4.1.3.1 CPU scheduling

The CPU queue is made up of as many as 15 subqueues. Each subqueue represents a band or group. The subqueues are maintained in descending order with the highest band at the top and the lowest band at the bottom of the CPU queue. User tasks with a high band value are given use of the CPU before any task with a lower CPU band value.

When a user uses its time slice, a new time slice is computed and the user task is moved to the bottom of CPU subqueue for that band.

4.1.3.2 Memory scheduling

The MRQ is a band ordered queue made up of subqueues. One subqueue represents a band and is priority ordered.

A job will stay in memory until one of the following two sets of conditions occurs:

1. A job with a higher band value is present in the MRQ, or a job of equivalent band value with a higher JXFMP. Also, the job has been in memory at least as long as the in-memory thrash lock.

2. A job with a higher memory band value is attempting an increase in field length and there is not enough free memory to satisfy the request. JSH checks if the request can be completed by allocating memory belonging to jobs with a lower memory band value to the requesting job. If JSH must roll jobs with a lower memory band value, in-memory thrash locks are ignored.

A job in memory will stay in memory until the in-memory thrash lock expires, providing that there are no jobs with a higher memory band value waiting for that memory space. A job rolled out stays rolled for the rest of the time specified by the out-of-memory thrash lock. The site analyst can adjust the size of the thrash lock value (see section 4.2, Installation Parameters).

4.1.3.3 Memory priority

A job's memory band value does not change unless an operatory changes the job priority.

4.1.3.4 Expanding job size

One exception to the memory priority scheme is JSH when a job in memory tries to expand. If there is not sufficient memory available for expansion, either in place or by relocating the job to a larger free memory segment, JSH tries to satisfy the request without a rollout. JSH computes the amount of memory of jobs in lower bands and compares it to the job attempting to expand. If the request can be satisfied by rolling out jobs from lower bands, JSH does so and then completes the expand request. If JSH could not satisfy the expand request, the job is rolled out regardless of its memory priority or thrash lock.

4.2 INSTALLATION PARAMETERS

This subsection discusses the COS installation parameters that have a bearing on JSH tuning. The following is a list of the JSH tuning parameters:

I@EXPANS	I@JSLK3	I@JSTS1
I@IJTL	I@JSLK4	I@JSTS2
I@JFLMAX	I@JSMPA	I@JSTS3
I@JOBMIN	I@JSMPB	I@JXTSIZ
I@JSCOS	I@JSMPC	I@MAXPAD
I@JSHTLE	I@JSMPD	I@MAXNUT
I@JSITS	I@JSRRI	I@MINPAD
I@JSLK1	I@JSTEI	I@NTXT
I@JSLK2	I@JSTS0	I@TSMIN

With the exception of I@TSMIN, all parameters associated with a time interval (seconds, milliseconds, or microseconds) are specified as arguments in $CYCLES macro. The $CYCLES macro translates a unit of time into the equivalent number of CPU cycles. (See the Macros and Opdefs Reference Manual, publication SR-0012, for details on $CYCLES.)

The following is a list of the installation parameters that are maintained as 64-bit data items in the STP tables area of COS:

I@JSCOS	I@JSLK3	I@JSMPD	I@JSTS2
I@JSHTLE	I@JSLK4	I@JSRRI	I@JSTS3
I@JSITS	I@JSMPA	I@JSTEI	I@MAXNUT
I@JSLK1	I@JSMPB	I@JSTS0	I@TSMIN
I@JSLK2	I@JSMPC	I@JSTS1	

Consequently, the system can be tuned while it is running by using the memory entry commands described in section 6, Station Debug Commands. Reassembly and restart of COS is not necessary until the site analyst arrives at a final parameter selection.

4.2.1 MAXIMUM NUMBER OF JXT ENTRIES (I@JXTSIZ)

I@JXTSIZ defines the number of Job Execution Table (JXT) entries and is the maximum number of jobs that can be active, as follows:

Name	Location	Default	Range
I@JXTSIZ	COSI@P	15 entries	1 through 256

4.2.2 MAXIMUM FIELD LENGTH (I@JFLMAX)

I@JFLMAX is the maximum amount of memory that a job can use, excluding the Job Table Area (JTA), as follows:

Name	Location	Default	Range
I@JFLMAX	COSI@P	77777_8 blocks	Up to machine size

4.2.3 INITIAL SIZE OF THE JOB TABLE AREA (I@IJTL)

I@IJTL is the initial size of the JTA. This quantity must be a multiple of 1000_8 and large enough for the fixed portion of the JTA plus three Dataset Name Tables (DNTs) as follows:

Name	Location	Default	Range
I@IJTL	COSI@P	10000_8	Not less than 7000_8

4.2.4 JOB TIME LIMIT EXTENSION (I@JSHTLE)

When a job first encounters a time limit, I@JSHTLE seconds are given to the job before exit processing begins. If a job reaches its time limit a second time, the job is aborted without the benefit of further exit processing. The I@JSHTLE parameter has the following values:

Name	Location	Default	Range
I@JSHTLE	STPTAB	3 seconds	Discretionary

4.2.5 PRECISION OF JOB DELAY INTERVAL (I@JSTEI)

Jobs entering a delay state or an event wait state in response to a J$AWAIT or J$DELAY function are placed in a queue and checked every I@JSTEI seconds for completion of the event or the delay interval. EXEC invokes the JSH at least once per second independent of the I@JSTEI value. As I@JSTEI increases, the accuracy of the interval enforced for a J$DELAY decreases. Similarly, the time when a job is resumed from a J$AWAIT request can differ from the time the event actually occurs by as much as the I@JSTEI value. This parameter has the following values:

Name	Location	Default	Range
I@JSTEI	COSI@P	1 second	See the preceding text

4.2.6 DISK ERROR DURING ROLLOUT (I@JSRRI)

When a disk error occurs while a job is rolling out, a message is issued to the master operator station display and the roll is retried in I@JSRRI seconds. This parameter has the following values:

Name	Location	Default	Range
I@JSRRI	COSI@P	20 seconds	Discretionary

4.2.7 IN-MEMORY THRASH LOCK (I@JSLK1 AND I@JSLK4)

I@JSLK1 and I@JSLK4 are the components of the in-memory thrash lock. After a job is placed in memory, the job is not eligible to roll out until I@JSLK1+T*I@JSLK4/1000 (where T is the time it takes to roll the job) seconds have elapsed. At that time, the job rolls out only if a job with a higher memory priority is waiting for memory. See subsection 4.3.2, Adjusting Memory Scheduling, for the range of allowable values. To increase the in-memory thrash lock by 1 second for each second of roll-in time, set I@JSLK4 to 1000. These parameters have the following values:

Name	Location	Default	Range
I@JSLK1	COSI@P	2 seconds	See preceding text
I@JSLK4	COSI@P	0 (integer)	See preceding text

4.2.8 OUT-OF-MEMORY THRASH LOCK (I@JSLK2 AND I@JSLK3)

I@JSLK2 and I@JSLK3 are the components of the out-of-memory thrash lock. The out-of-memory thrash lock is determined from the formula TL=I@JSLK2+T*I@JSLK3/1000, where TL is the thrash lock. After a job is rolled out, the job is not eligible for a memory allocation until I@JSLK2+T*I@JSLK3/1000 (where T is the roll-in time) seconds have elapsed. At that time, the job is placed into the memory request queue and competes for memory along with other jobs. If I@JSLK3 is nonzero, the out-of-memory thrash lock is proportional to the size of the job. See subsection 4.3.2, Adjusting Memory Scheduling, for the range of allowable values. To increase the out-of-memory thrash lock by 1 second for each second of roll-out time, set I@JSLK3 to 1000. These parameters have the following values:

Name	Location	Default	Range
I@JSLK2	COSI@P	4 seconds (in cycles)	See preceding text
I@JSLK3	COSI@P	0 (integer)	See preceding text

4.2.9 INITIAL TIME SLICE (I@JSITS)

I@JSITS is called the initial time slice. Whenever a job is granted a memory allocation, the first time slice for each of its tasks is the value I@JSITS, which should not be larger than the smallest time slice. If I@JSITS is set to a large value without adjusting memory scheduling parameters and there are many jobs in memory, some jobs may not execute after gaining a memory allocation. This parameter has the following characteristics:

Name	Location	Default	Range
I@JSITS	COSI@P	0.1 seconds	Less than 1 second

4.2.10 DISCONNECT COST (I@JSCOS)

I@JSCOS is the disconnect cost. Whenever a user task is disconnected from the CPU, I@JSCOS is subtracted from the user task's time slice. The purpose of this action is to prevent I/O-bound user tasks from sitting at the top of the CPU queue for long periods without using any CPU time. Increasing I@JSCOS has the effect of increasing the rate at which I/O-bound user tasks use their time slices and thereby the accuracy of the job's memory priority. I@JSCOS should be kept small unless the site has a problem with I/O-bound user tasks dominating the system. This parameter has the following characteristics:

Name	Location	Default	Range
I@JSCOS	COSI@P	25 microseconds	Discretionary

4.2.11 MINIMUM TIME SLICE (I@TSMIN)

I@TSMIN is the minimum time slice. Before JSH requests EXEC to connect a user task to the CPU, JSH ensures that the user task to be connected has a time slice of at least I@TSMIN. This action prevents the situation where the expense (system overhead) of connecting and disconnecting a user task exceeds the resources delivered to the user task. The minimum value must be larger than 1 millisecond. This parameter has the following characteristics:

Name	Location	Default	Range
I@JSMIN	STPTAB	3 milliseconds	Greater than 1 millisecond

4.2.12 TIME SLICE (I@JSTS0, I@JSTS1, I@JSTS2, AND I@JSTS3)

A new time slice is calculated whenever a user task's time slice expires and the user task is still eligible for execution. Use the following formula when calculating the time slice:

TS=I@JSTS3*P**3+I@JSTS2*P**2+I@JSTS1*P+I@JSTS0

(TS is the time slice in seconds and P is the job statement priority). The following default values are assigned to the coefficients making up the time slice formula:

Name	Location	Default	Range
I@JSTS0	COSI@P	0.3 seconds	See subsection 4.3.2, Adjusting Memory Scheduling
I@JSTS1	COSI@P	0.1 seconds	
I@JSTS2	COSI@P	0.0 seconds	
I@JSTS3	COSI@P	0.0 seconds	

Use the following general rules when determining time slice duration:

- If the number of jobs in memory is large, small time slices guarantee an equitable distribution of the CPU.

- If time slices are set to large values, be sure to consider lengthening the memory residence times. A job could gain a memory allocation, execute for I@JSITS seconds, and not get the CPU again without exceeding its privilege to stay in memory.

4.2.13 MEMORY PRIORITY (I@JSMPA, I@JSMPB, I@JSMPC, AND I@JSMPD)

Use the following formula to determine the memory priority of a job in memory. MP is the memory priority, P is the job statement priority, CJS is the current job size in words, and TIM is the time in memory (wall-clock time since last roll in). I@JFLMAX is the maximum job size.

```
MP=MAX[P-I@JSMPD,P+I@JSMPA+I@JSMPB*CJS/I@JFLMAX-TIM/I@JSMPC]
```

The following default values are assigned to the coefficients in the preceding memory priority formula:

Name	Location	Default	Range
I@JSMPA	COSI@P	1 priority unit	See subsection 4.3.2,
I@JSMPB	COSI@P	0	Adjusting Memory
I@JSMPC	COSI@P	10 seconds	Scheduling
I@JSMPD	COSI@P	0.1 priority unit	

4.2.13.1 Calculating initial priority

To calculate the initial memory priority given to a job when it is rolled in, use the following term:

```
P+I@JSMPA+I@JSMPB*CJS/I@JFLMAX
```

Use the term for initial memory priority only once.

4.2.13.2 Memory priority aging

Memory priority is aged by evaluating the term TIM/I@JSMPC each time the job relinquishes the CPU (disconnects) and then subtracting this value from the initial memory priority. When the computed priority falls below P-I@JSMPD, MP stops decreasing and remains constant at P-I@JSMPD (through action of the MAX function).

The parameter I@JSMPC (called slope) determines how quickly memory priority decays. Figure 4-1 is a graphic representation of memory priority (MP).

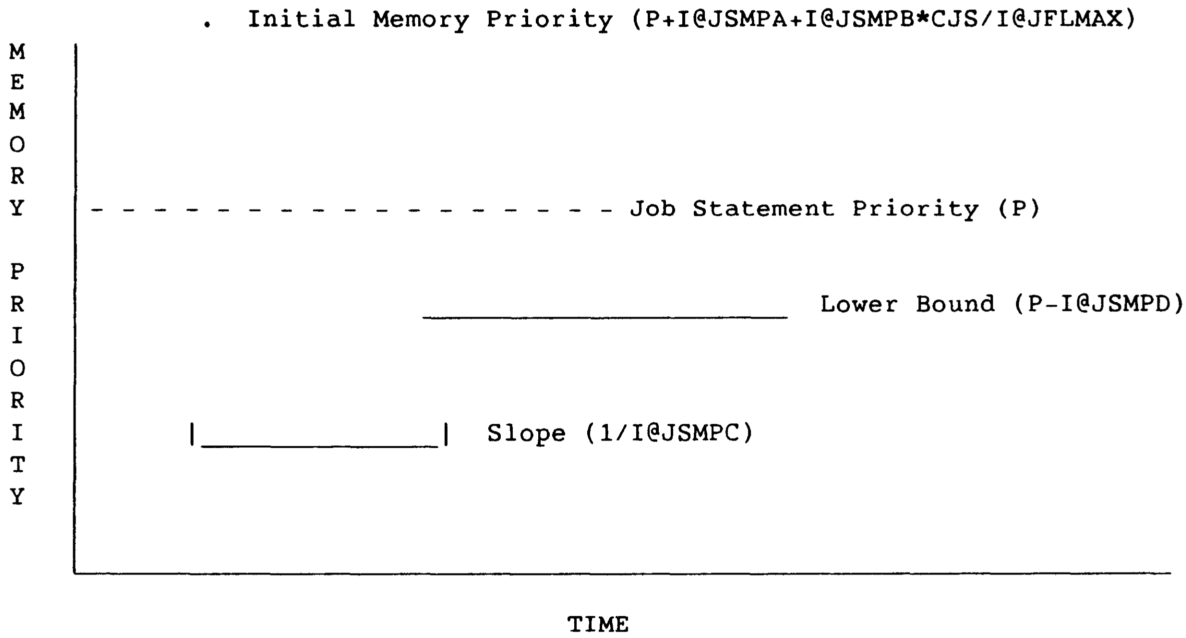

Figure 4-1. Memory Priority

4.2.14 EXPANSION SPACE IN MEMORY (I@EXPANS AND I@JOBMIN)

When a job is waiting for memory, JSH does not make an allocation unless I@EXPANS words in memory are left free after the allocation. The memory left free need not be contiguous with the current allocation. The requirement is ignored if there are fewer than I@JOBMIN jobs in memory. These parameters give the ability to leave a quantity of memory readily available for expansion space. These parameters have the following characteristics:

Name	Location	Default	Range
I@EXPANS	STPTAB	50000_8 words	Discretionary
I@JOBMIN	STPTAB	2 jobs	Discretionary

4.2.15 USER FIELD LENGTH INCREMENT (I@MINPAD AND I@MAXPAD)

If a job requests additional memory, the system guarantees that the memory increase is at least I@MINPAD words. If a user job attempts to relinquish memory, the system keeps the relinquished memory in the pad area of the job until the pad area exceeds the I@MAXPAD words. These parameters have the following characteristics:

Name	Location	Default	Range
I@MINPAD	USERI@P	4000_8 words	Discretionary
I@MAXPAD	USERI@P	10000_8 words	Discretionary

4.2.16 USER TASKS ALLOWED (I@NTXT AND I@MAXNUT)

I@NTXT defines the maximum number of Task Execution Table (TXT) entries and is the maximum number of user tasks that can be active. I@MAXNUT is the maximum number of user tasks allowed to be active in any single user job. These parameters have the following characteristics:

Name	Location	Default	Range
I@NTXT	COSI@P	I@JXTSIZ	I@JXTSIZ through 256
I@MAXNUT	STPTAB	1 user task per job	1 through I@NTXT

4.2.17 JOB PRIORITY BAND TABLE

The mapping of job priorities into bands is defined with the Job Priority Band Table (JPB). The JPB is defined in STPTAB. Sixteen entries make up the JPB, and each entry has 3 variables; CPU, memory, and I/O. It is recommended that you define all 16 entries (priorities 0 through 15).

The following is the format for the JPB table:

Format:

Location	Result	Operand
	BUILD	JP,(LE@JPB),(IO=*xx*,MEM=D'*yy*,CPU=D'*zz*) P *nn*

xx I/O band value
yy Memory band value
zz CPU band value
nn Job card priority

Figure 4-2 shows the default definition for the JPB table.

```
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  0
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  1
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  2
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  3
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  4
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  5
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  6
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  7
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  8
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  9
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P 10
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P 11
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P 12
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P 13
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P 14
BUILD      JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P 15
```

Figure 4-2. Default Definition of JPB Table

4.3 TUNING THE JSH

In tuning JSH, installation parameters are adjusted so that the scheduling processes meet the objectives of site administration. Before the site analyst sets out to tune JSH, scheduling objectives must be determined. The site analyst should address two questions:

1. How are the machine resources to be distributed over the anticipated load?

   Usually, this question is answered by classifying the jobs that constitute the load into job classes and assigning a priority to each class using the JCSDEF utility (see the Operational Aids Reference Manual).

2. How responsive should the JSH be?

The site analyst must then decide how much the scheduler discriminates between jobs of different priority to meet the objectives. When time slices and thrash locks are short and memory priority aging is rapid, JSH is called responsive. A responsive scheduler is costly in the amount of system overhead required for the additional work done by EXEC and by JSH. Clearly, responsiveness is necessary in an interactive environment but not in most batch environments. Figure 4-2 shows tuning JSH.

Once the load is determined and classified, CPU priorities are adjusted, memory priorities are adjusted, and system performance is observed. If performance is acceptable, the process is complete. Otherwise, priorities are readjusted until performance is satisfactory.

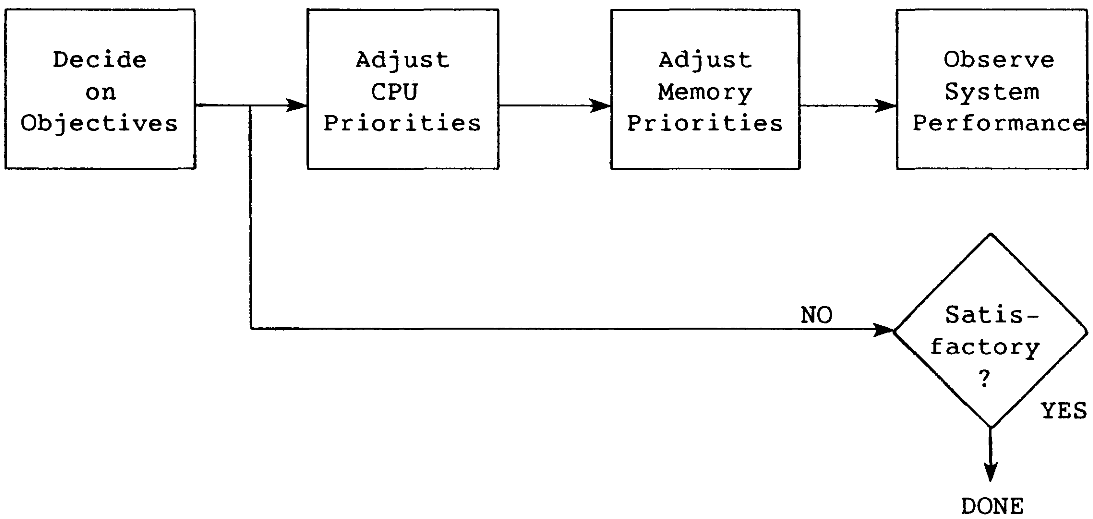

Figure 4-3. Tuning the Job Scheduler

4.3.1 ADJUSTING CPU SCHEDULING

Adjusting the duration of the time slice is the major way of controlling CPU priority. The four parameters I@JSTS0, I@JSTS1, I@JSTS2, and I@JSTS3 can be set to values so almost any distribution of time slice can be obtained.

Figures 4-4, 4-5, and 4-6 show three jobs as examples of the STATUS display after three CPU-bound jobs have executed together in memory for approximately 100 wall-clock seconds. These jobs are identical in all respects with the exception of their priorities (7, 9, and 11 for JOB1, JOB2, and JOB3, respectively).

These examples illustrate that CPU time is distributed proportionately among jobs in memory. Figure 4-4 uses the time slice parameter settings of the released system. A notable quality of figure 4-4 is that the difference in the amount of time accumulated by a low and a high priority job is not very large. The difference is much more noticeable in figures 4-5 and 4-6 where the high-order terms of the time slice formula have nonzero coefficients.

CRAY SYSTEM STATUS

CSDN = DAY-27

QUEUES E I O R S FRAME 0

JSQ TID	DC	DATASET	CLASS	STATUS	PRI	USED	TIME LIMIT	FIELD LENGTH	ID
1518	IN	JOB3	LARGE	WAIT-CPU	11.0	39	*****	196	II
1516	IN	JOB1	LARGE	WAIT-CPU	7.0	27	*****	196	II
1517	IN	JOB2	LARGE	EXECUTE	9.0	33	*****	196	II

END OF DATA

Figure 4-4. Default Time Slice Parameters

The STATUS display shown in figure 4-4 was generated by assigning the indicated values to the following parameters:

		JOB1	JOB2	JOB3
I@JSTS0 = 0.3 seconds	priority	7	9	11
I@JSTS1 = 0.1 seconds	time slice	1.0	1.2	1.4
I@JSTS2 = 0.0 seconds	theoretical			
I@JSTS3 = 0.0 seconds	distribution	27.7	33.3	38.8

CRAY SYSTEM STATUS

CSDN = DAY-27

QUEUES E I O R S FRAME 0

JSQ	DC	DATASET	CLASS	STATUS	PRI	TIME USED	TIME LIMIT	FIELD LENGTH	ID	TID
1535	IN	JOB1	LARGE	WAIT-CPU	7.0	18	*****	196	II	
1536	IN	JOB2	LARGE	EXECUTE	9.0	30	*****	196	II	
1537	IN	JOB3	LARGE	WAIT-CPU	11.0	48	*****	196	II	

END OF DATA

Figure 4-5. Time Slice Parameters with Nonzero High-order Terms, Example 1

The STATUS display shown in figure 4-5 was generated by assigning the indicated values to the following parameters:

			JOB1	JOB2	JOB3
I@JSTS0 = 0.1	seconds	priority	7	9	11
I@JSTS1 = 0.0	seconds	time slice	1.57	2.53	3.73
I@JSTS2 = 0.03	seconds	theoretical			
I@JSTS3 = 0.0	seconds	distribution	20.0	32.3	47.6

CRAY SYSTEM STATUS

CSDN = DAY-27

QUEUES E I O R S FRAME 0

JSQ	DC	DATASET	CLASS	STATUS	PRI	TIME USED	TIME LIMIT	FIELD LENGTH	ID	TID
1535	IN	JOB1	LARGE	WAIT-CPU	7.0	21	*****	196	II	
1536	IN	JOB2	LARGE	WAIT-CPU	9.0	32	*****	196	II	
1537	IN	JOB3	LARGE	EXECUTE	11.0	45	*****	196	II	

END OF DATA

Figure 4-6. Time Slice Parameters with Nonzero High-order Terms, Example 2

The STATUS display shown in figure 4-6 was generated by assigning the indicated values to the following parameters:

				JOB1	JOB2	JOB3
I@JSTS0 =	0.1	seconds	priority	7	9	11
I@JSTS1 =	0.1	seconds	time slice	1.63	2.54	3.74
I@JSTS2 =	0.01	seconds	theoretical			
I@JSTS3 =	0.001	seconds	distribution	20.6	32.1	47.3

4.3.1.1 Theoretical time slice distribution

The theoretical distribution referred to in figures 4-4 through 4-6 is the amount of time each job should accrue if system overhead and operator timing errors are disregarded. You can calculate the theoretical distribution as follows.

Assume that TS(7), TS(9), and TS(11) are the time slices of jobs with priority 7, 9, and 11, respectively. If these three jobs are in memory together for an interval (I), a job of priority 7 accrues time as in the following proportion:

```
TS(7)
---------------------- * I
TS(7)+TS(9)+TS(11)
```

4.3.1.2 Determining values for time slice parameters

Two factors should be considered when deciding on the values for the time slice installation parameters:

- Average length of time a job spends in memory
- Average number of jobs in memory

Determining the first factor, average memory residence time, is discussed later in this section. The second factor, number of jobs in memory, is determined by observing the STATION display showing executing jobs.

The main objective in setting the time slice is to achieve a proportional distribution of CPU time among the jobs in memory. Avoid the situation where a job rolls in, executes for the initial time slice (I@JSITS), and rolls out without getting any more CPU time. This occurs when the time slice is large compared to the average stay in memory. If the number of jobs in memory is large, the time slice must be small. If the average number of jobs in memory is small, the size of the time slice can be large without sacrificing accuracy in the distribution of CPU time. If time slices are set to large values (greater than 2 seconds), the accuracy of the distribution of CPU time can be maintained by increasing the amount of time the average job spends in memory.

4.3.2 ADJUSTING MEMORY SCHEDULING

Three aspects of memory priority can be controlled by the site analyst:

- Initial boost at roll in
- Rate at which priority ages
- Lower bound

4.3.2.1 Memory priority example

Assume that two jobs have been entered into the system (Job A at priority 10 and Job B at priority 5), and both jobs do not fit into memory at the same time. As the tuning parameters are released, the following sequence of events occurs:

1. Job A initiates and gains a memory allocation before Job B.

2. When Job A gets into memory, its memory priority is boosted to 11 (P+I@JSMPA+I@JSMPB*CJS/I@JFLMAX=10+1+0=11). See figure 4-7, point A.

3. As Job A runs, it is disconnected at the end of each time slice and its memory priority is updated (aged).

4. After I@JSLK1+(T*I@JSLK4/1000) seconds, the thrash lock expires and the job is eligible to roll but does not because Job A has a higher memory priority than Job B.

5. After 10 seconds (released value of I@JSMPC), Job A's memory priority ages back down to 10.

6. Shortly after that, Job A's memory priority reaches its lower bound (10-0.1=9.9) and remains constant. See figure 4-7, point B.

Job B does not run until Job A is suspended or terminates. Assume that Job A is suspended for a request to the user Exchange Processor (EXP). As the tuning parameters are released, the following sequence of events occurs:

1. Since the job is suspended and Job B is waiting, Job A rolls out immediately.

2. As soon as memory is released, Job B rolls in and begins executing (figure 4-7, point C).

3. Assume that Job A is resumed. After I@JSLK1+(T*I@JSLK4/1000) seconds, Job B is eligible to roll.

4. Its priority is now slightly less than 6 (P+I@JSMPA+CJS*I@JSMPB/I@JFLMAX-I@JSLK1/I@JSMPC=5+1+0-0.2=5.8).

5. After I@JSLK2+(T*I@JSLK3/1000) seconds, the thrash lock expires for Job A and it is moved to the MRQ with priority 10.

6. As soon as Job B is disconnected at the end of its time slice, its memory priority is found to be less than Job A at the top of the MRQ and Job B immediately begins to roll out (figure 4-7, point D).

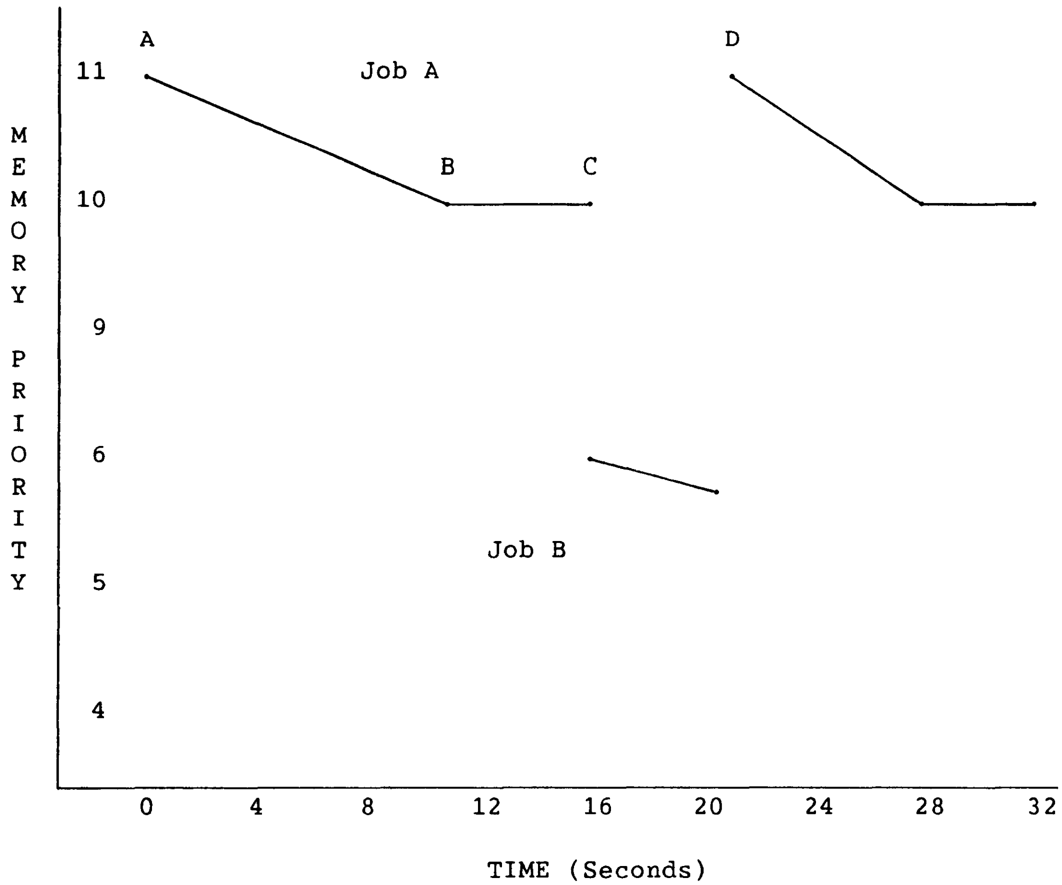

Figure 4-7. Memory Priority, Example 1

In this situation, the slope and the initial boost have no effect on the precedence of Job A over Job B. Job A is always in memory as long as job B is the only job waiting. If, in the course of executing, Job A is suspended for any reason, Job B gets a chance to use the CPU. When Job B succeeds in getting a memory allocation, its residence is not controlled by slope, initial boost, or lower bound, but by the thrash locks. The amount of time that Job B spends in memory is based entirely on the number of times that Job A is suspended.

Figure 4-8 demonstrates Job B executing for 4-second intervals because the out-of-memory thrash lock (I@JSLK2) ensures no memory contention for that length of time.

If priority 10 jobs are not to share the system with priority 5 jobs except for reasons of efficiency, the default parameter settings need not be changed. Conversely, changing the lower bound (I@JSMPD), so that a priority 10 job ages to something less than priority 5, guarantees that priority 5 jobs get into memory for a short time (I@JSLK1=2 seconds) at least once every 60 seconds. (It takes 60 seconds for a priority 11 job to age to priority 5.)

To obtain still more interaction, two things can be done:

- The in-memory thrash lock (I@JSLK1) can be increased.
- The slope can be increased.

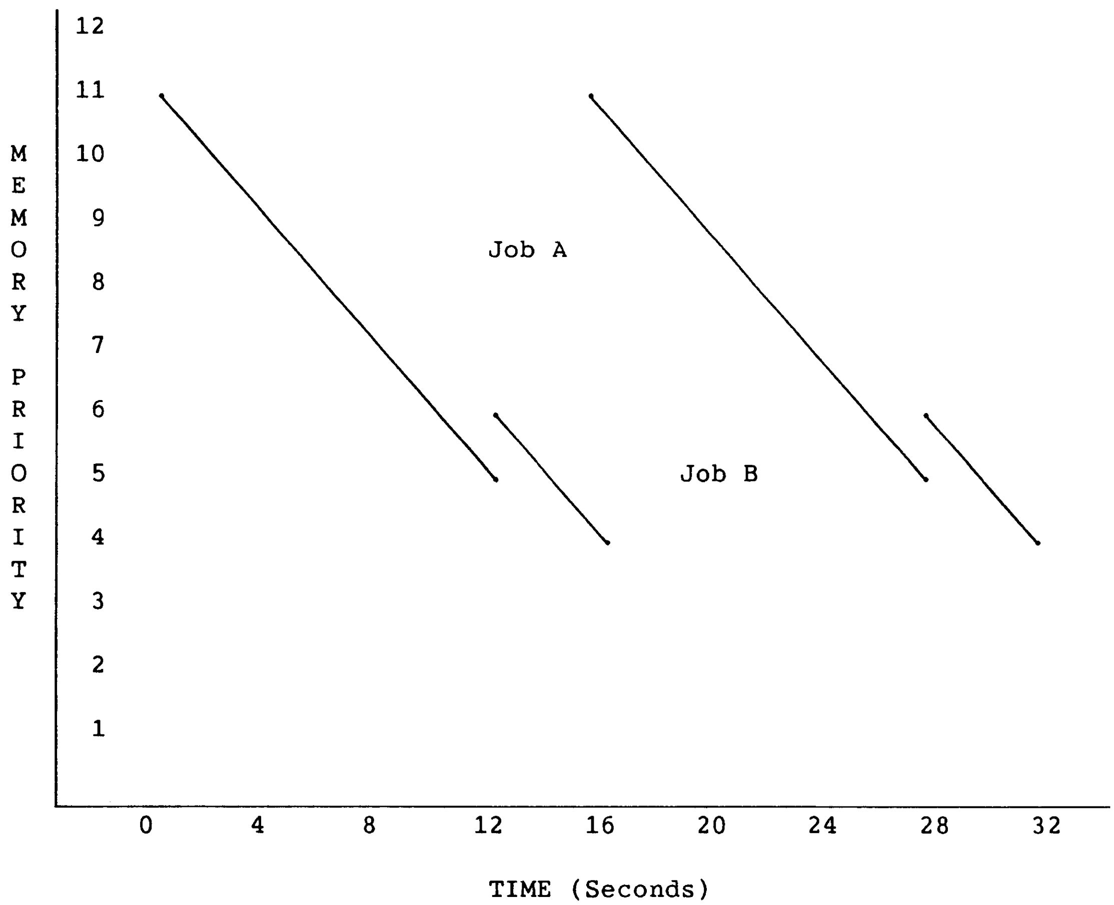

Figure 4-8. Memory Priority, Example 2

If I@JSLK1 is increased to 3 seconds, priority 5 jobs are eligible to execute 3 out of every 63 seconds or about 5 percent (3/(60+3)) of the time (disregarding the job size factor). If slope is changed to 2 seconds/priority unit, Job B is in memory every 12 seconds (time to age from 11 to 5) and with I@JSLK1 set to 2 seconds, is eligible to execute 2 out of every 14 seconds or 14 percent (2/(12+2)) of the time. See figure 4-8.

4.3.2.2 Cray computer system example 1

Consider a more realistic example involving a 2-million-word CRAY-1 S computer with four tape drives. The site administrator determines that jobs at the site can be classified in three groups:

- Small jobs (T<20, MFL<D'131000)
- Tape jobs of any size
- All other jobs

Classes are defined for these three groups named SMALL, TAPE, and NORMAL. The maximum amount of memory any job can use is approximately half of memory (I@JFLMAX=3400_8 sectors). The following scheduling objectives are used:

- Small jobs have little impact on the system if their numbers are limited to three (20 percent of memory) and should get prompt service.
- Demand for tape drives is fairly high. To maintain efficient drive use, tape jobs should be favored over normal jobs.
- Large jobs (in either the TAPE or NORMAL classes) should stay in memory longer than small jobs to compensate for the added system overhead of rolling a large job.

The bulk of the load on this system is likely to fall in the classes mentioned before, but exceptions like very high or very low priority jobs are sure to arise. The class priorities are located in the middle of the range 0 through 15.

Priority 8, 7, and 6 are assigned to the SMALL, TAPE, and NORMAL classes, respectively. The first objective is satisfied because SMALL jobs have the highest priority. A waiting SMALL job always gains a memory allocation before a waiting NORMAL or TAPE job. Since small jobs can never use more than 20 percent of memory, their impact is minimal. The second objective is satisfied because tape jobs have a higher priority than normal jobs.

To realize the third objective, the coefficients in the memory priority formula must be changed from their released values. Since memory priority must be proportional to job size, I@JSMPB is set to a nonzero value. Because I@JSMPB is nonzero, I@JSMPA may have to be changed (reduced) to keep memory priority from being too large or the slope may have to be increased to cause aging to occur faster. Picking values for the installation parameters in the memory priority formula is done by making an initial selection, visualizing the priority curve (possibly by plotting the data), and then refining the original estimates.

Picking I@JSMPB=1.0 as the initial selection means that for every million words of memory a job uses, one priority unit is added to the job's memory priority. Since no job can use more than about 1 million words, the largest value of CJS*I@JSMPB/I@HFKMAX is 1. The largest job in the TAPE class would then get a memory priority boost of 2. Such a job would take 10 seconds to age within range of a waiting SMALL job and would never interact with a NORMAL job.

Changing I@JSMPD, the lower bound, to 1.1 allows SMALL jobs to interact with TAPE jobs and TAPE jobs to interact with NORMAL jobs. Thus, a large TAPE job (P=7) ages to priority 6 after being in memory for 30 seconds. Similarly, a large NORMAL job enters memory with priority 8 and ages to priority 7 after 10 seconds. Assuming contention from only a large TAPE job and no SMALL jobs, a large NORMAL job is in memory 10 out of every 40 seconds or 25 percent of the time. If I@JSMPA is reduced from 1 to 0.5, large NORMAL jobs would be in memory 5 out of every 30 seconds or 16 percent of the time.

Selecting the distribution of time slices for this example is straightforward. Assume that no changes are made to the coefficients in the time slice formula. Time slices are then 0.9, 1, and 1.1 seconds for NORMAL, TAPE, and SMALL jobs, respectively. If three SMALL jobs are entered into the system along with several TAPE and NORMAL jobs, the three SMALL jobs are in memory almost all the time. The CPU is distributed among the jobs according to their time slices.

If only one or two TAPE and NORMAL jobs fit into memory with the SMALL jobs, the small jobs get roughly 60 to 75 percent of the available CPU time. A steady flow of SMALL and very large TAPE and NORMAL jobs maintains this situation. The only reason for associating SMALL jobs with priority 8 is to guarantee that they always get into memory and receive prompt service.

Changing the time slice distribution to something similar to figure 4-5 or 4-6 only gives a higher percentage of the CPU to SMALL jobs. This situation can be avoided by doing one of two things:

- Decrease the number of SMALL jobs allowed to run at one time
- Change the criteria which determine a SMALL job (for example, T<12)

4.3.2.3 Cray computer system example 2

This example shows another site with the following constraints:

- A set of users have real time constraints. A job(s) has to get through the machine as fast as possible.
- Another set of users need speedy throughput but not exclusive rights (express)
- The CPU on-line diagnostics are to run in background mode. CPU's are assigned to the on-line diagnostics only in the absence of other work (OLDIAGS).
- Users that have consumed more than their resource allotments for a month will execute only when there are no jobs belonging to users who are still within their assigned resource allotments. Site administrators monitor machine resource usage per user, and assign users that have consumed their resource allotments to a background class. The background class has the effect of still allowing the consumed user to access the machine, but not impacting the turnaround of the users that stay within their resource allotments. (BACKGROUND)

JSH can be tuned to realize the above constraints using bands.

Jobs will receive the following assignments:

Priority 1 for on-line diagnostics
Priority 2 through 5 for background jobs
Priority 6 through 10 for normal jobs
Priority 11 for express jobs
Priority 12 through 15 for exclusive rights jobs

The band table for this configuration is shown in figure 4-9.

```
BUILD        JP,(LH@JPB ),(TN='JPB'
BUILD        JP,(LE@JPB ),(IO=D'00,MEM=D'00,CPU=D'00  P  0
BUILD        JP,(LE@JPB ),(IO=D'00,MEM=D'05,CPU=D'00  P  1
BUILD        JP,(LE@JPB ),(IO=D'01,MEM=D'01,CPU=D'01  P  2
BUILD        JP,(LE@JPB ),(IO=D'01,MEM=D'01,CPU=D'01  P  3
BUILD        JP,(LE@JPB ),(IO=D'01,MEM=D'01,CPU=D'01  P  4
BUILD        JP,(LE@JPB ),(IO=D'01,MEM=D'01,CPU=D'01  P  5
BUILD        JP,(LE@JPB ),(IO=D'02,MEM=D'02,CPU=D'02  P  6
BUILD        JP,(LE@JPB ),(IO=D'02,MEM=D'02,CPU=D'02  P  7
BUILD        JP,(LE@JPB ),(IO=D'02,MEM=D'02,CPU=D'02  P  8
BUILD        JP,(LE@JPB ),(IO=D'02,MEM=D'02,CPU=D'02  P  9
BUILD        JP,(LE@JPB ),(IO=D'02,MEM=D'02,CPU=D'02  P 10
BUILD        JP,(LE@JPB ),(IO=D'03,MEM=D'03,CPU=D'03  P 11
BUILD        JP,(LE@JPB ),(IO=D'04,MEM=D'04,CPU=D'04  P 12
BUILD        JP,(LE@JPB ),(IO=D'04,MEM=D'04,CPU=D'04  P 13
BUILD        JP,(LE@JPB ),(IO=D'04,MEM=D'04,CPU=D'04  P 14
BUILD        JP,(LE@JPB ),(IO=D'04,MEM=D'04,CPU=D'04  P 15
```

Figure 4-9. Example Definition of JPB Table

4.4 OBSERVING JSH

The best way to observe JSH performance is to watch the STATUS display with a short refresh rate (using the REFRESH,ON,1 operator command for Data General sites and REFRESH,ON,.1 operator command for IOS sites). The relevant fields on the STAT display are the priority, status, field length, and time used with status being the primary piece of information. You may also use the DSPL station command to view many of JSHs internal queues. Refer to subsection 6.4, Special IOS Station Displays, for a description of the DSPL station command.

Verifying that the scheduler is working properly is difficult. The STATUS display status field is limited to a single piece of information. When a job rolls out, for example, R-OUT is used to describe the state of the job. The job could have rolled for any of several different reasons. Similarly, a job can stay rolled out for several reasons other than simple memory contention.

If a job rolls out or stays rolled out and there appears to be no reason, check the JXT in memory using the memory displays in section 6. The second word (word 1) of the job's JXT entry contains the ASCII status characters associated with the job. The COS Internal Reference Manual, Volume II: STP, publication SM-0141, defines the meaning of these characters.

5. COS OPERATIONS

This section is directed to the on-site analysts and development programmer/analysts setting up procedures for the operators and handling unusual situations for which the operators are not prepared. This section assumes the software has been installed on the CRAY Y-MP, CRAY X-MP, CRAY X-MP EA, or CRAY-1 computer system as described in section 3, COS Installation, and in the System Installation Bulletin.

5.1 SYSTEM POWER-ON/POWER-OFF CONSIDERATIONS

This subsection describes some of the effects of power cycles on the execution of COS and some procedures to avoid problems caused by the power cycles. It does not describe the actual power-on/power-off procedure.

After the computer system has been powered on and before COS is started, all front end computers should be powered on, but the channels should be turned off. If a front end cannot be powered on, then the interface box must be powered on off-line. If the front end does not have an interface box, the cables must be disconnected from the front end and connected to themselves in a loopback. If the front end is connected but not powered on when COS is started, COS receives a steady interrupt on that channel and can do nothing but process the interrupt.

Before powering down the computer system, COS should be idled down with the SHUTDOWN command to suspend all jobs and the CHANNEL OFF command to prevent new input. The system log buffer should then be flushed by running EXTRACT with the FLUSH option; otherwise, the part of the system log still in memory is lost when the mainframe is powered down.

After the log buffer has been flushed, the Central Processing Unit (CPU) should be stopped. For example, the following command can be issued from a Data General maintenance control unit (MCU) without an I/O Subsystem (IOS):

```
DUMP D 0 1
```

This causes 1 word to be dumped to a scratch file and stops the CPUs. On systems with an IOS, the CPUs are stopped by setting word 200_8 in memory to a nonzero value.

The following command can be issued at an IOS station console:

```
200=1,E
```

The disks should be put off-line before powering off so that power variations do not send erroneous signals to the disk.

5.2 STARTUP PARAMETER FILE

A parameter file must be available to the station connected to the MCU channel (and thus able to perform maintenance control functions) for COS to be started through a STARTUP command issued at that station.

The parameter file contains one record per directive, is generated by the operator or an analyst, and must be available to the station. The STARTUP command assumes by default that the name of this file is COSPAR although normally another file is specified.

The parameter file is a set of directives, each consisting of a key word (except for disk flaws), and arguments as required by the key word. Arguments are separated from each other and from the key word by commas. Except for *asciidata*, the first space encountered is considered the start of a comment and ends the effective portion of the parameter record. Each parameter file record, including any comment portion, is terminated by an ASCII return code, 015_8. In the following formats, italic type represents variable data to be added, and square brackets enclose optional arguments. The startup file is in unblocked format and must be created by the IOS file editor.

5.2.1 STARTUP MODE DIRECTIVES

Only one Startup mode directive can be selected and each directive selects a Startup mode. The last directive encountered is used. These directives have no arguments.

Formats:

```
*INSTALL
*DEADSTART
*RESTART
```

If one of the above directives is not present, the Startup mode is determined by an installation parameter I@ZOPT, which signals restart by default.

For *INSTALL, COS is started as if for the first time. All CRAY Y-MP, CRAY X-MP, CRAY X-MP EA, or CRAY-1 mass storage is assumed to be vacant, and the dataset catalog (DSC) and device labels are initialized.

For *DEADSTART, COS is started as if after a normal system termination.

For *RESTART, COS is started as if after a system interruption. For a *RESTART only, jobs that were in execution at the time of the interruption can be recovered from their latest roll image at the operator's option.

5.2.2 ENTER MEMORY DIRECTIVES

Memory directives enter a value into a specified memory word or parcel within a word. The parameter file can contain any number of the following directives, presented in the forms noted. If two or more directives change the same parcel, the last one encountered is implemented. Memory directives are EXEC relative.

To enter numeric data into memory, use the following directive:

Format:

```
*EMEM,addr,data[,startbit[,bitcount]]
```

addr Absolute address. It may be either a word address or a parcel address. If a parcel, the parcel specifier may be a through d, in either uppercase or lowercase.

data Value to put in the specified address. If *addr* is a word address, value may be up to 22 octal digits; if *addr* is a parcel address, value may be up to 6 octal digits.

startbit Optional starting bit in words. If *startbit* is present, *addr* must be a word address. *startbit* is a decimal number in the range 0 through 63. The default is 0.

bitcount Optional number of bits beginning at *startbit*. If *bitcount* is present, *startbit* must also be present (it can be 0). *bitcount* is a decimal number in the range 0 through 63. If *bitcount* is 0, or is not supplied, it is taken to be from *startbit* to bit 63 of the word.

data is always stored right-justified within the bit range or parcel, and must be a valid octal number. If *data* exceeds the range specified by *startbit* and *bitcount*, it is truncated without comment. If *data* exceeds 22 characters, it is an error, and the directive is ignored and a message is issued to the master console when STARTUP subroutine Z begins execution.

To enter ASCII text into memory, use the following directive:

Format:

```
*ENASCII,addr,value[,startbit[,bitcount]][,R]
```

addr Absolute memory address. *addr* must be a word address.

value Data to be placed in the specified word. *value* need not be enclosed in quotes unless it contains one of the characters recognized as separators for COS control statements (blank, comma, left or right parenthesis, colon, caret, period, quotation marks, and equal sign). If the data contains quotation marks, each must be doubled.

startbit Optional starting bit in the word. *startbit* is a decimal number in the range 0 through 63. Default is 0.

bitcount Optional number of bits beginning at *startbit*. If *bitcount* is present, *startbit* must also be present (it can be 0). *bitcount* is a decimal number in the range 0 through 63. If *bitcount* is 0, or not supplied, it is taken to be from *startbit* to bit 63 of the word.

R Optional right-justification flag. By default, ASCII data is left-justified in the word or bit field, and blank-filled on the right within the field. If R is present, the data is right-justified, with blank fill on the left within the field.

value is always truncated to fit into the field without comment. A message is issued by STARTUP subroutine Z if *value* exceeds 8 characters.

5.2.3 BREAKPOINT SELECTION DIRECTIVES

Three directives set breakpoints during system startup: *EBP, *DEBUG, and *BRE. The *DEBUG command is specifically intended for use in debugging Startup.

The *EBP command sets a breakpoint in any portion of STP, including Startup. This command is the older form of breakpoint entry.

Format:

```
 ______________________________________
|                                      |
| *EBP,n,addr1,parcel1[,addr2,parcel2] |
|______________________________________|
```

n Breakpoint number, in the range 0 through 7

addr1,addr2
Word address relative to STP

parcel1,parcel2
Parcel indicators. Values must be in the range a through d, in uppercase or lowercase.

addr2 and *parcel2* are optional, but specify the breakpoint reset address. If *addr2* is present, *parcel2* must also be present. *parcel2* must not be present if *addr2* is not present.

*DEBUG allows Startup to calculate and set a breakpoint automatically. This breakpoint is set at the entry point to the specified Startup option (Install, Deadstart, or Restart) and may be used to halt Startup without requiring changes to the parameter file each time the system is reassembled. The S registers contain an ASCII message informing the operator that the breakpoint was set by a *DEBUG command rather than parameter file errors.

Format:

```
 ________
|        |
| *DEBUG |
|________|
```

The *DEBUG command performs the same function as the following command, where *nnnnnp* equals the address of ZINSTALL, ZDEAD, OR ZRESTART:

*EBP,1,*nnnnn,p*

*EBP and *DEBUG may be used in the same parameter file, but if *DEBUG is present, breakpoint number 1 cannot be used on an *EBP directive within the same parameter file.

The *BRE directive performs the same function as the *EBP directive, using syntax more similar to that of the debug BREAKPOINT command.

Format:

```
*BRE,number,addr[,resetaddr]
```

number Breakpoint number, in the range 0 through 7

addr Parcel address at which the breakpoint is to be set

resetaddr Optional parcel address at which the breakpoint is reset at *addr*

When breakpoints are placed in Startup, some special problems arise. Since breakpointed code must be restarted by commands from the operator station, the breakpoint must occur after SCP has been initialized. It is then possible to log on while Startup is waiting at a breakpoint. When Startup is at a breakpoint, only the debug functions should be used.

5.2.4 *END DIRECTIVE

The *END directive signals the end of the parameter file. It has no parameters. If no *END is present, directives are processed until an end-of-file is reached.

Format:

```
*END
```

5.2.5 MEMORY DIRECTIVES

The *MEMSIZE directive is used to reduce the amount of physical main memory available to COS. It overrides, but cannot exceed the amount specified by the C@MMSIZE parameter.

COS is limited to running in 16 Mwords of memory or less. On machines with more than 16 Mwords of memory, a *MEMSIZ value above 16 Mwords becomes a no-op.

Format:

```
 _______________
|               |
| *MEMSIZ,size  |
|_______________|
```

size Memory size in octal words

The *UPPER and *LOWER directives are used on CRAY X-MP mainframes when COS is configured with both halves of memory, but the operator wishes to temporarily change the mainframe switch to physically configure only the upper or lower half of memory. This information is used by memory error correction and reporting to correct for the difference between the physical memory configuration and the assembled COS memory configuration. This parameter only affects CRAY X-MP systems, and by default is both halves.

Format:

```
 _________
|         |
| *UPPER  |
|_________|
```

or

```
 _________
|         |
| *LOWER  |
|_________|
```

5.2.6 DEVICE DIRECTIVES

Device directives provide the capability of changing a site's configuration at Startup time. These directives allow the operator to modify fields in the Equipment Table (EQT), the Tape Device Table (TDT), the Channel Configuration Table (CNT), and the Link Configuration Table (LCT) (that is, they allow you to add or delete devices from the current configuration or access existing devices through other channels). Device directives can also be used to delete permanent datasets residing on a device or to retain a dataset but flag it as inaccessible.

5.2.6.1 *RESTORE directive

The *RESTORE directive causes the information on a fast secondary storage (FSS) device (SSD or Extended Buffer Memory) to be restored from a backup dataset. This directive is useful when the FSS device will be power-cycled but the datasets on the device need to be preserved. The operator makes a backup copy using the FLUSH command after doing a shutdown. The *RESTORE directive is used on the subsequent startup to restore the datasets.

Format:

```
*RESTORE,ldv
```

ldv Logical device name of the device for which the back-up information is to be restored

Startup allocates a backup dataset for each FSS device defined as a volatile in the EQT. See the VOL parameter for mass storage devices in subsection 5.2.6.2. Also see the VOL parameter on the EQT macro in subsection 2.11.6.

5.2.6.2 Device control

The operator can define or delete tape devices as well as change the attributes of existing tape devices with the *CONFIG parameter file directive. The operator can also define or delete mass storage devices defined in the EQT and control CPU and High-speed (HSP) or Very-high-speed (VHSP) Channel availability.

Format (tapes):

```
*CONFIG,DVN=dvn[,RDONLY][,NAVAIL][,GDN=gdn][,UN=un][,DT=dt]
                 RDWRT     AVAIL

  [,ICHn=iopchan:cuid][,IOP=iopid][,SYSTEM]
                                    MAINT
```

DVN=*dvn*	Tape device name (1 to 8 ASCII characters)
RDONLY	Device is in read-only mode.
RDWRT	Data can be read or written.
NAVAIL	Device is not present or cannot be used.
AVAIL	Device is present and can be used.
GDN=*gdn*	Generic device name (1 to 7 ASCII characters). The default generic device name is *TAPE, a dual-density tape device. See section 2 for a definition of *generic name*.
UN=*un*	Unit identifier (decimal digits in the range of 0 through 15)
DT=*dt*	Device capability descriptor. The permissible values are:

GCPE@200	6250/1600 b/i dual-density tape drive, 200 ips
IBM@3480	IBM 3480 magnetic tape

ICH*n*=*iopchan*:*cuid*

I/O Processor (IOP) channel number descriptor. The key word is ICH1, ICH2, ICH3, or ICH4 depending on the channel to which the directive refers. One through four IOP channels can be associated with each device.

iopchan	IOP channel number where device is attached
cuid	One or two control units connected to the specified IOP channel through which the device can be accessed (must be a single hexadecimal digit, 0 through 9 or A through F).

IOP=*iopid*	IOP identifier; 0 through 3.
SYSTEM	Returns the device to the system from maintenance mode
MAINT	Places device in maintenance mode. Only diagnostics will be allowed access to the device.

NOTE

The GDN field in the CNT must match one of the generic device names defined for the system.

If no parameters other than the device name are given, the Configuration Table (CNT) is searched for the named device. The device entry is deleted if found; if not found, the command is ignored without comment. If additional parameters are given but the device does not exist in the CNT, the device is added and a Tape Device Table (TDT) entry is built for it. If the device does exist and additional parameters are given, the CNT and TDT entries are updated according to the parameters supplied.

When the *CONFIG directive is used for mass storage device control, it must contain the DVN=*ldv* parameter. All other parameters are optional.

Format (mass storage):

```
*CONFIG,DVN=ldv[,RDONLY][,NAVAIL][,GDN=gdn][,UN=unit][,DT=dt]
                 RDWRT    AVAIL

  [,SCX=Y][,IOP=iop][,CPT=cpt][,CH=ch1][,MSD=Y][,RBN=Y][,SCR=Y]
        N                                    N       N       N

  [,VOL=Y][,WDL=Y][,CTL=Y][,RLS=Y][,GRP=n]
        N       N       N       N

  [,STK=stk][,NTK=ntk][,SYSTEM]
                        MAINT
```

DVN=*ldv* Logical device name of the DD-19, DD-29, DD-39, DD-40, DD-49, BMR, or SSD mass storage device, field EQLDV.

RDONLY Device is in read-only mode.

RDWRT Data can be read or written.

NAVAIL Device is not present or cannot be used.

AVAIL Device is present and can be used.

GDN=*gdn* Generic device name

UN=*unit* Primary unit number, field EQUNT

DT=*dt* Device type to be stored in EQDT. *dt* may be one of the following:

DD19	DD-19 disk drive
DD19I	DD-19 with cylinders 1 through 41_8 reserved for IOS diagnostics

DD29	DD-29 disk drive
DD29I	DD-29 with cylinders 1 through 41_8 reserved for IOS diagnostics
DD39	DD-39 disk drive
DD39I	DD-39 with cylinders 1 through 61_8 reserved for IOS diagnostics
DD40	DD-40 disk drive
DD40I	DD-40 with cylinders 1 through 21_8 reserved for IOS diagnostics
DD49	DD-49 disk drive
DD49I	DD-49 with cylinders 1 through 23_8 reserved for IOS diagnostics
EBM	Buffer Memory device type
SD8	8-million-word SSD
SD16	16-million-word SSD
SD32	32-million-word SSD
SD64	64-million-word SSD
SD128	128-million-word SSD
SD256	256-million-word SSD
SD512	512-million-word SSD

SCX=Y / N — SCX indicates that the device is to be viewed by the software as residing on its own channel. The parameter values and their definitions follow:

- Y Sets the EQSCX field
- N Clears the EQSCX field

IOP=*iop* — IOP number to which the device is connected, field EQIOP

CPT=*cpt* — Control path type indicating the location and type of device driver associated with a device CPT. The following is a list of values and their definitions:

- DCU DCU mainframe/disk
- IOS IOS-IOS/disk
- SSD SSD-mainframe/SSD

CH=ch_1 — Primary channel number, field EQCHN

MSD=Y / N — Defines whether *ldv* is the master device as follows:

- Y Sets the EQMSD field
- N Clears the EQMSD field

RBN=Y
N

Defines whether the device must be requested by name. Not all devices can be requested by name. At least one device must be public. The codes and their descriptions that you may use follow:

Y Sets the EQRBN field
N Clears the EQRBN field

SCR=Y
N

Device *ldv* is a scratch device; *ldv* is not a scratch device. Permanent datasets may not be saved on a scratch device if parameter I@PERM = 0, in COSI@P. A scratch device may be selected with DV=LDV, or ST=SCR at assign time. A scratch device may also be selected if I@STYPE = 1 in COSI@P. A job that is recovered across a restart is allowed to save a local dataset on a device that had not been previously declared scratch, regardless of the current SCR setting.

VOL=Y
N

Device *ldv* is volatile; *ldv* is not volatile. Space is reserved on a nonvolatile device at startup time to copy the entire volatile device when requested using operator command FLUSH. *ldv* must be either an SSD or EBM device name.

WDL=Y
N

Y sets EQUP, causing Startup to write the device label if a valid label is not found. The parameter WDL=N causes the EQUP field to be cleared.

NOTE

If a volatile device has been powered off, be sure to write a device label (WDL=Y) when the device is returned to COS, because Startup expects all mass storage devices to have a device label.

CTL=Y
N

Device *ldv* is a controlled device; *ldv* is not controlled. A controlled device must have its space reserved before assigning files to it.

RLS=Y
N

Y sets EQRLS, causing Startup to release all permanent datasets residing wholly or partially on this device. RLS=N clears the EQRLS field.

GRP=*n*

Disk device group identifier. GRP identifies the device as a member of a disk group (stripe) or deletes a member of a group. The master device cannot be defined as a member of a group.

GRP=*n* (continued)	*n* is a decimal digit, 0 through 9. If *n* is 0, the device is deleted from its group, if any. If *n* is nonzero, it specifies the *id* of the group to which the device belongs. If the device already belongs to another group or is not a member of any group, the device is added to the specified group and deleted from any previously specified group. Device groups are supported only for IOS-resident devices.
STK=*stk*	Starting track number for a logical device in octal. Used when defining more than one logical device on a physical device. The Master Device cannot be defined on a logical device unless that device starts on track zero. Default is to start at physical track 0.
NTK=*ntk*	The number of tracks present on the logical device in octal. Must be specified if STK is specified. The value of STK + NTK must not exceed the maximum number of tracks available on the device type defined. Default is to use the full device capacity. Table 2-1 shows the maximum number of tracks available for each mass storage device. Refer to this table for more information.
SYSTEM	Returns the device to the system from maintenance mode
MAINT	Places device in maintenance mode.

Format (CPUs):

```
*CONFIG,DVN=dvn,CPU=cpun[,DOWN][,SYSTEM]
                          UP      MAINT
```

DVN=*dvn*	Name of the CPU entry in the CNT
CPU=*cpun*	CPU number (0 through *n*). At least one CPU must remain UP at all times.
DOWN	The CPU is not scheduled to user or system tasks.
UP	The CPU becomes eligible for scheduling to user or system tasks.

SYSTEM Returns the CPU to the system from maintenance mode

MAINT Places CPU in maintenance mode. Only diagnostics will be allowed to run in this CPU.

Format (HSP or VHSP channels):

```
*CONFIG,DVN=dvn,CH[=ch:ch][,ON ]
                             OFF
```

DVN=*dvn* Logical name of the SSD device, field EQLDV

CH=*ch:ch* Channel numbers. If CH appears without arguments, all channels to the SSD are affected; otherwise, only the specified channels are affected.

ON Makes channel available

OFF Makes channel unavailable

If all configured channels to the SSD are changed from ON to OFF, the SSD will automatically be changed to NAVAIL in the EQT. Likewise, if at least one of the channels is turned ON, the SSD will be changed to AVAIL. (IF more than one logical SSD has been defined using the EQT macro STK and NTK parameters, all logical SSDs will be made AVAIL or NAVAIL.)

A directive of the following form causes the corresponding EQT or TDT entry to be deleted:

*CONFIG,DVN=*ldv*

If a *CONFIG directive is used that has no corresponding entry in the EQT, a new entry is built. When specifying a new EQT, the DT=*dt* parameter must also be specified.

If any parameters are not specified, the values currently in those fields are not changed.

Example Startup parameter files:

Assume that disk device DD-19-30 is configured and has been available. Due to hardware problems, DD-19-30 must be removed from the system. Later, the hardware problems are corrected, and DD-19-30 is put back into use. The following steps show the commands used to solve the problem:

1. File before hardware problem:

```
*RESTART
.
.
.
*END
```

2a. File to make a device unavailable:

```
*RESTART
*CONFIG,DVN=DD-19-30,NAVAIL
.
.
.
*END
```

Datasets allocated to the device remain in the Dataset Catalog (DSC) but cannot be accessed by users.

2b. File to make a device unavailable and delete permanent datasets:

```
*RESTART
*CONFIG,DVN=DD-19-30,NAVAIL,RLS=Y
.
.
.

*END
```

Datasets allocated to the device are deleted from the DSC. This format should be used if the hardware problem is such that data errors are likely (head crash, scratched pack, and so on).

2c. File to leave device available for reading:

```
*RESTART
*CONFIG,DVN=DD-19-30,RDONLY
.
.
.
*END
```

This format allows the operator to attempt to dump datasets that reside on the device. Normally, this format is followed by a startup using the parameter file in example 2b to delete the datasets after they have been dumped.

3. File to restore DD-19-30:

If the data on the pack is still valid, the device can be restored simply by removing the *CONFIG parameter file entry. During a subsequent startup, the DSC flags for datasets on the device are cleared to allow access from user jobs.

If the data on the pack is believed to be invalid or if the device label cannot be found, the pack can be relabeled and restored to the system by the following parameter file.

```
*RESTART
*CONFIG,DVN=DD-19-30,AVAIL,WDL=Y,RLS=Y
.
.
.
*END
```

The RLS parameter deletes any datasets that reside on DD-19-30. WDL=Y allows Startup to relabel the device. RLS=Y should be removed before the next startup (when RLS is set to N, it is not necessary to remove the RLS parameter).

5.2.6.3 Link configuration control

The Link Configuration Table (LCT) can be modified at Startup time. The *LCT directive provides the ability to define or change existing entries.

Format:

```
*LCT,ENT=xx[,CHO=co][,OFF][,CHT=ct][,CHN=ch]
                      ON
```

ENT=*xx* Ordinal position of the LCT entry in the table, starting with one

CHO=*co* Channel ordinal, field LCCHO

ON/OFF Sets/clears the LCON field

CHT=*ct* Channel type, field LCCHT can equal one of the four following types:

IFC	Channel coupler
NSC	NSC HYPERchannel
VAX	VAX version of IFC, A side
VBX	VAX version of IFC, B side
GST	Guest operating system station

CHN=*ch* Valid channel pair number, in decimal

More than one *LCT command can be used per LCT entry and only the specified information is changed; all other information in the LCT entry remains the same.

5.2.7 DISK FLAW DIRECTIVES

Disk flaw directives reserve disk tracks with flaws or reserve tracks for engineering diagnostic use. The directives must appear in sequence, and once the sequence begins, no other directive is valid until the sequence ends. An invalid directive in the sequence is treated as an unrecognized key word.

Format (for reserving disk tracks):

```
*FLAW,ldv
flaw directives
*ENDFLW
```

ldv The logical device to which all subsequent flaw cards apply until an *ENDFLW directive is encountered

flaw directives
One or more specifications in any of the following forms identifying the disk areas to be reserved:

C*nnn*	Flaw all of cylinder *nnn*
C*nnn-mmm*	Flaw cylinders *nnn* through *mmm*
C*nnn*,T*mm*	Flaw track *mm* of cylinder *nnn*
C*nnn*,T*mm-pp*	Flaw track *mm* through track *pp* of cylinder *nnn*

Flaws for a device can be in any order. All numbers are octal. Normal processing of directives resumes following the *ENDFLW directive. These directives provide for adding flaws encountered during system operation without requiring reassembly of the system. The addition of a flaw prevents users from accessing a permanent dataset that may occupy a flawed track. Further use of such a dataset requires the dataset to be recreated after the flaw is added. Note that if flaws are added that are outside the boundaries of the logical device, they will be ignored without comment (see the EQT macro parameters STK and NTK in subsection 2.11.6, Equipment Table Macro (EQT))

If the parameter file contains any *FLAW directives, the flaw information in the device label is rewritten at the end of the startup. This requires a flaw to be entered into the parameter file only once to be permanently recorded. Once the system has been started using a parameter file containing a new flaw, that flaw may be removed from the parameter file for subsequent startups. (The exception to this rule is the label track. If the label track is to be flawed, the *FLAW card must be left in the parameter file until the EFT can be updated to reflect the flaw.) Another sequence of directives allows deletion of a previously specified flaw. The same restrictions apply to the placement of these directives as to the *FLAW directive sequence.

Format (for deleting a flaw):

```
*DELFLAW,ldv
flaw directives
*ENDFLW
```

ldv Logical device where all subsequent flaw cards apply until an *ENDFLW directive is encountered

flaw directives
One or more specifications in any of the following forms identifying the disk areas to be reserved:

C*nnn*	Delete all flaws from cylinder *nnn*
C*nnn-mmm*	Delete flaws from cylinders *nnn* through *mmm*
C*nnn*,T*mm*	Delete flaws from track *mm* of cylinder *nnn*
C*nnn*,T*mm-pp*	Delete flaws from track *mm* through track *pp* of cylinder *nnn*

Any flaws deleted through the *DELFLAW directive are released in the Disk Reservation Table (DRT), and the device label is rewritten at the end of the startup process. The first occurrence of a *DELFLAW directive permanently removes from the device label any flaw that was initially specified through a *FLAW parameter file directive; only one startup parameter file need be changed for flaws specified in this manner.

If the original specification of the flaw is by the Engineering Flaw Table (EFT), the flaw is reinstated at the next startup, since Startup never rewrites the EFT block. To permanently remove such flaws, each copy of the parameter file must contain a *DELFLAW directive for the flaw. If flaws are specified that are beyond the boundaries of the logical device, they will be ignored without comment (see the EQT macro parameters STK and NTK in subsection 2.11.6, Equipment Table (EQT) Macro).

The *SKIPEFT directive controls startup action if no EFT is found on a device. If this directive is present in the parameter file, Startup continues without comment if it cannot find EFT data. If the directive is not present and the device type is other than SSD or EBM, Startup notifies the operator of the missing EFT with a message at the master operator station and waits for a response (this message occurs for each device that does not have an EFT). If an EFT is present on all devices, this directive has no effect. (The SSD and BMR devices are special in that they normally do not have an EFT. STARTUP therefore will not notify the operator of missing EFTs on these devices, even if no *SKIPEFT command is present.) Also, the operator will be notified only once per physical disk, even if more than one logical device is defined on it.

Format:

```
 ___________
|           |
| *SKIPEFT  |
|___________|
```

Parameter deck example:

```
*DEADSTART
*CONFIG,DVN=DD-19-30,AVAIL
*CONFIG,DVN=DD-19-31,AVAIL
*FLAW,DD-19-30
C27,T4
C37,T4-6
*ENDFLW
*FLAW,DD-19-31
C307
C312-314
C403,T2
*ENDFLW
*CONFIG,DVN=DD-19-32,AVAIL,WDL=Y
*FLAW,DD-19-32
C12,T11
*ENDFLW
*CONFIG,DVN=DD-19-33,NAVAIL,RLS=Y
*END
```

5.2.8 DUMP CONTROL DIRECTIVES

Two directives control system dump processing during Startup: *DUMP and *NODUMP. *DUMP forces Startup to create a saved dataset containing the image of the preallocated system dump area. *NODUMP inhibits Startup from processing the system dump. It is flagged as an error if both directives appear in the same parameter file.

When a system is installed, a system dump area is allocated on disk. The first sector of the allocated area contains a list of allocation units set aside for the system dump.

Ordinarily, when a system dump is written to the preallocated area of a disk, a flag is set in the first sector of the dataset indicating that a new (uncopied) system dump exists. During the next restart or deadstart, this flag is recognized, and a copy of the preallocated area is made and saved as a permanent dataset. After successfully saving the copy of the dump, Startup rewrites the sector to clear the flag so that subsequent startups do not copy the same dump again. If the user wishes to make another copy of the dump or if the flag is improperly set during the dump process, *DUMP may be used to cause Startup to unconditionally copy and save the dataset even though the flag is clear in the preallocated area.

Format:

```
 _________
|         |
| *DUMP   |
|_________|
```

The presence of a *NODUMP line in the parameter file causes Startup to clear the flag that indicates a dump is present. Disk space is still reserved in the DRT, but the dump will not be copied.

Format:

```
 ___________
|           |
| *NODUMP   |
|___________|
```

5.2.9 ROLLED JOB RECOVERY CONTROL DIRECTIVES

When restart is selected, the operator can also select recovery of rolled jobs. Recovery cannot be selected for install or deadstart. If an attempt is made to select recovery for either deadstart or install, the directive is ignored and a message sent to the system log indicates that recovery was not possible.

Recovery is selected by the rolled job recovery parameter.

Format:

```
 __________
|          |
| *RRJ,n   |
|__________|
```

n The level of recovery to be performed. Two levels are supported. If *n*=1, recovery is attempted; if *n*=0, no recovery is performed.

The *LOCK command tells Startup what action to take with regard to jobs being recovered when the system presently being started is not the same as the one in use when the current roll image was created. In some cases, most notably when a job was rolled out to await response to an ACQUIRE, FETCH, or DISPOSE statement, the job is in a state in which it cannot be safely resumed on a different system. This problem occurs with jobs containing one or more tasks for which the TCEPC field is set.

*RJJ only works if the JXT, DNT and RJ Table sizes have not changed. If any of these tables have been changed, no jobs can be recovered.

Another case is when the amount of memory available for a user job decreases from the amount available in the previously running system. Jobs requesting too much space cannot be run. In these cases, the operator can instruct Startup how to treat the job, using the *LOCK command. *LOCK only works if the JXT, DNT and RJ Table sizes have not changed. If any of these tables have been changed, no jobs can be recovered. The default is set by the I@LOCK installation parameter (see appendix A) at system generation time.

Format:

```
 ___________
|           |
| *LOCK,n   |
|___________|
```

n Represents the option desired. The values and meanings for *n* are:

0 Process the job normally, as if TCEPC were not set. This allows the job to be recovered in progress and to be resumed upon completion of Startup. If this option is used, make sure EXP addresses are not changed.

n (continued)	1	Recover the job and reserve all necessary resources, but do not allow the job to execute until another restart is performed using the proper system. A flag is set in the Job Execution Table (JXT) for the recovered job. The Job Scheduler (JSH) does not consider the job as a candidate for roll in. The status display shows a status of L-SYS for jobs with TCEPC set and L-MEM for jobs requesting too much memory.
	2	Do not recover the job. Rerun it from the beginning if possible.

Only jobs with the TCEPC field set or jobs requesting more than the maximum memory are affected by these options.

5.2.10 SYSTEM DIRECTORY RECOVERY DIRECTIVE

Whenever you select restart or deadstart, the operator can request that the System Directory (SDR) not be recovered by specifying the *SDR directive. A new edition of $SDR is allocated, and any datasets to be in the SDR must be reentered when this directive is specified. If the SDR recovery directive is not specified, the operating system recovers the resident SDR by processing records from the $SDR dataset.

Format:

```
*SDR
```

5.2.11 JOB CLASS STRUCTURE DIRECTIVE

The operator can invoke a new job class structure when either deadstart or restart without recovery of rolled jobs is selected, using the job class structure directive.

Format:

*JCLASS,PDN=*pdn*[,ID=*userid*][,ED=*edition*][,OWN=*owner*][,RD=*read*]

PDN=*pdn* Name of the permanent dataset containing the job class structure definition to be invoked. If the dataset does not exist or cannot be read, an appropriate system log message is issued, and the default job class structure goes into effect.

ID=*userid* User ID for the permanent dataset

ED=*edition*
Permanent dataset edition number. If this parameter is not specified, the highest edition is used.

OWN=*own* Owner identification for the permanent dataset (optional)

RD=*read* Read permission control word, if needed

5.2.12 SUBMIT STARTUP JOB DIRECTIVE

For either a deadstart or a restart, the operator can cause jobs to be submitted as part of the startup procedure. While STARTUP jobs exist, no other jobs are allowed to begin execution or are accepted from the stations. STARTUP-submitted jobs, by definition, are not recovered across a restart.

Format:

```
*SUBMIT,PDN=pdn[,ID=userid][,ED=edition][,OWN=owner][,RD=read]
```

PDN=*pdn* Permanent dataset name of job to be submitted by STARTUP; a required parameter. Cannot exceed 7 characters.

ID=*userid* User ID for the permanent dataset (optional, default is none)

ED=*edition*
Permanent dataset edition number (optional, default is highest edition number)

OWN=*owner* Owner identification of the permanent dataset (optional, default is SYSTEM)

RD=*read* Read permission control word (optional)

To enable STARTUP submitted jobs, there must be a class defined in the Job Class Structure Definition Table for them. The parameter I@ZJCLN defines the name of this class.

5.2.13 $SYSTEMLOG BUFFER SUPPRESSION DIRECTIVE

During a startup, the operator can suppress the recovery of the $SYSTEMLOG buffer in memory by including the *SUPSYS directive in the parameter file. A flag set in Startup tells the Log Manager to suppress buffer recovery. After the system log is successfully accessed during initialization, the system log is read until end-of-file (EOF) is reached, and the high memory table is initialized. This procedure prevents messages previously residing in the buffer from being recovered.

Format:

```
*SUPSYS
```

5.2.14 FORCE $SYSTEMLOG EDITION DIRECTIVE

During a startup, the operator may include the *SYSLOG directive in the parameter file to force a new edition of the system log. A flag set in Startup tells the Log Manager to save a new edition. The Log Manager checks the flag during initialization of the system log and, if it is set, saves either the edition specified by the operator or the next highest edition if none is specified.

Format:

*SYSLOG[,*ed*]

ed $SYSTEMLOG number of the edition to be used. The default is the next highest edition. If specified, the edition must be greater than the current highest edition of $SYSTEMLOG.

5.2.15 DXT CREATION AND SIZE CHANGE

The $DSC-Extension dataset (DXT) is created automatically if it does not already exist and if startup is not a restart. Startup will not complete unless the DXT exists.

5.2.15.1 Creation

The *DXT directive and/or installation parameters control where the DXT resides on disk. The default installation parameter values cause the DXT to be created on the master device with no overflow allowed. The default does not guarantee contiguous disk space for the DXT. The installation parameter values can be adjusted during Startup through the *DXT directive.

Format:

```
*DXT,SZ=size,DV=ldv,OVF=opt,CAI=opt
```

SZ=*size* Size of DXT; decimal number of 512-word blocks. If *size* is prefixed with a +, the DXT size is increased the specified number of blocks. If *size* is unsigned and the DXT already exists, this parameter is ignored.

DV=*ldv* Logical device name indicating where the DXT is to begin. This parameter is ignored if the DXT size is being increased.

OVF=*opt* Overflow option; *opt* can be either yes (Y) or no (N).

CAI=*opt* Contiguous AIs option; *opt* can be either Y or N.

All parameters on the *DXT directive are optional. That is, if the directive consists of simply *DXT, the installation parameter values I@DXTSZ, I@DXTLDV, I@DXTOVF, and I@DXTCAI are used. After the DXT has been created, Startup completes the initialization of the STP-assembled DNT and Device Allocation Table (DAT) for the DXT dataset and the DXT allocation bit map XAT in high memory.

5.2.15.2 Size change

You can change the size of the DXT dataset after it has been created by using the *DXT directive in the Startup parameter file. For example:

```
*DXT,SZ=+100
```

This *DXT example increases the size of the DXT by 100_{10} 512-word blocks. The additional disk space allocated to the DXT is initialized to all zeros except for the Block Control Words (BCWs).

5.2.16 INSTALLATION PARAMETER MODIFICATION DIRECTIVE

The *IPARM command provides the ability to change certain installation parameters. Currently, the only installation parameter that can be changed by this command is I@IOPICH (see appendix A).

Format:

```
*IPARM,sym=value
```

sym Installation parameter name minus the I@. The only legal value is IOPICH.

value Value to be assigned to installation parameter *sym*

5.2.17 TIME-STAMP CONVERSION DIRECTIVES

Time-stamps are one-word representations of a date/time combination. The DSC entry for each permanent and I/O dataset contains a number of time-stamps for recording the creation, alteration, last access, and last dump times.

Different versions of the way time-stamps are encoded make it necessary for Startup to perform a conversion to whichever time-stamp system is in use. The *TSCONV directive controls this conversion. This directive applies only if one or more datasets were created for a COS version prior to 1.13 and does not apply to any new installations.

Format:

```
*TSCONV,ON
        OFF
```

ON Default condition. If ON is specified or defaulted, STARTUP converts all time-stamps in the DSC to the representation appropriate for the system being run. This option is also used by the Permanent Dataset Manager (PDM) when creating DSC entries for PDSLOAD; *TSCONV,ON causes all time-stamps entered into the DSC to match the system being run.

OFF STARTUP and PDM avoid conversion of any time-stamps.

Time-stamps created by ACCESS, ADJUST, DISPOSE, MODIFY, PDSDUMP, SUBMIT, SAVE, and system datasets are always generated as appropriate for the system in use, and are totally independent of the *TSCONV directive. Therefore, when *TSCONV,OFF is specified, the DSC might contain a mixture of time-stamp versions.

5.2.18 FAST SECONDARY STORAGE PREEMPTION DIRECTIVES

Fast Secondary Storage devices (FSSs), which are configured as generic resources, can be oversubscribed by COS. Oversubscription means that separate jobs whose total FSS needs are greater than the size of the device are allowed to initiate and share the device. Resource needs are indicated on the JOB command. Sharing is performed on a job basis. When a resource is oversubscribed, each job using the resource takes a turn with its FSS datasets resident on the device. When a job is not resident, it is suspended from execution.

High priority jobs are granted device access before low priority jobs. Jobs of equal priority share the device in a round-robin fashion. Operator commands SWEEP and RESTORE are used to manually control which jobs have device access. Refer to the I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051, for a description of the SWEEP and RESTORE commands. The FSS preemption directives are used to declare attributes of the preemption environment as follows:

1. Names of generic resources that can be oversubscribed.
2. The extent of oversubscription allowed by the system (oversubscription factor)
3. The minimum time that a job is guaranteed device access.
4. Availability of flush volatile device.
5. Devices preferred for holding preempted dataset images.
6. Swap space partitioning.

The oversubscription factor is an integer number of device images. Swap space is the volume of disk space needed to accommodate an oversubscription factor. The oversubscription factor is enforced by the operating system by controlling job initiation. The system refrains from initiating a job with FSS needs that will cause the oversubscription factor to be violated.

The system does nothing to guarantee that the volume of swap space implied by the oversubscription factor is available. The site analyst can specify which devices are preferred for use as swap space. Swap space is not isolated from use by the operating system or by user jobs unless request-by-name devices are specified as preferred devices. Users must be urged not to use these devices.

The operator is informed continuously as to the availability of swap space through the SWAP display. If the volume of local and permanent datasets grows so that the full volume of swap space is not available, swap space is reduced by the extent of the infringement. The infringement is divided among all preemptable devices. Swap space partition capacity is reduced, starting with the lowest priority partition, until the full infringement is reflected for each preemptable resource. This behavior results in a gradual shutdown of job initiation that begins with low priority jobs. When job initiation is limited due to lack of swap space, a warning is placed on the SWAP display and a mandatory response message is issued to the STM display at 5-minute intervals.

If swap space is depleted to a point where preemptable resources cannot be cleared of datasets with the operator SWEEP command, preemption is disabled and a warning is placed on the SWAP display with a mandatory response message at 5-minute intervals. Warnings on the SWAP display disappear when swap space becomes available.

Except for declaration of memory pool size (I@MP6SZ) in the site configuration mod, all attributes of the preemption environments are established through startup parameter file directives. If the system attempts to initiate a job needing preemptable resources and fails due to insufficient memory pool space, a warning is placed on the SWAP display along with a mandatory response message on the STM display at 5-minute intervals. Refer to the I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051, for a description of the SWAP display.

Formats:

```
*FSS
*PREEMPT,GRN=grn,OS=n,NOFLUSH
*DEVICE,LDV=ldv1:ldv2:...ldv8
*RESIDE,TL=n
*SPACE,LP=n,UP=m,P=p
*ENDFSS
```

The *FSS and *ENDFSS directives indicate the beginning and end of the FSS preemption directives.

The PREEMPT directive specifies a generic resource to be declared preemptable. A PREEMPT directive is entered for each preemptable generic resource. The parameters on the PREEMPT directive are the following:

GRN=*grn* Generic resource name associated with a controlled FSS device in the EQT (Buffer Memory or SSD). The device must be configured as a scratch device.

OS=*n* Oversubscription factor. *n* is an integer greater than zero and represents the number of device images allowed to accrue during COS operation. An oversubscription factor of 1 allows the resource to be swept by the operator but does not allow any sharing of the device (refer to the SWEEP command in the I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051). An oversubscription factor of 2 allows two device images to accrue and permits an oversubscription ration of 2:1.

NOFLUSH This option tells COS that no area should be preallocated for use by the FLUSH command. A flush area is always reserved for any volatile device in the EQT unless the device is configured preemptable and the NOFLUSH option is used.

The DEVICE directive names preferred devices for holding preempted datasets. If a DEVICE directive is not specified, preempted datasets are stored on the default disk devices. The parameter of the DEVICE directive is as follows:

LDV=*ldv1:ldv2:..ldv8*
Logical device name of any device in the EQT except a preemptable generic resource.

The RESIDE directive indicates the minimum residence time granted to a job when it gains an FSS allocation. Only operator intervention or job termination causes the device to be freed prior to the residence time. The parameter of the RESIDE directive is as follows:

TL=*n* Thrash lock. The minimum time, in minutes, that a job is allowed to have access to a preemptable generic resource. *n* must be an integer.

The SPACE directive is a way of reserving a percentage of swap space for jobs of a certain priority range. Swap space is allocated according to the following rules:

When a job needing resources enters the system, swap space is allocated from the partition associated with the priority of the job.

If space cannot be found in a partition of the same or lower priority, the job does not initiate.

Space is never allocated from partitions with a higher priority than the incoming job.

The priority of a job can be changed by the operator with the ENT jsq P n command. When a job needs preemptable resources, a priority change causes the system to look in a different partition for swap space. When a job is executing and already has a swap space allocation, a priority change can cause the allocation to violate the partitioning rules, particularly if a job's priority is reduced. On this occasion, the system attempts to realign the swap space allocation so that it adheres to the partitioning rules. This may not always be possible since a lower priority partition may be full. When a realignment cannot be performed, an indicator is set on the RSTAT display and the job continues executing. Jobs needing space from the partition occupied by the job must wait for the realignment to occur or for the job to terminate.

A maximum of three SPACE directives can be specified. No more than three partitions can be configured. The parameter of the space directives are as follows:

LP=*n* Lower priority. n is an integer between 1 and 14 (and must be less than UP) that indicates the minimum priority of a job obtaining swap space from this partition.

UP=*n* Upper priority. n is an integer (between 0 and 15) that indicates the maximum priority of a job obtaining swap space from this partition.

P=*p* Percentage. *p* is an integer between 1 and 100 that indicates the size of the partition as a percentage of the total swap space.

The LP and UP parameters must be specified so that only one undeclared partition is formed. If the percentage options on the SPACE directives do not total 100, the remaining percentage is attributed to the undeclared partition. If there is no undeclared partition and the percentages do not total 100, the remainder is attributed to the lowest priority partition.

5.2.19 RESOURCE DATASET MANAGEMENT (RDM) DIRECTIVES

There are three Resource Dataset Management directives: RDMRSD, RDMSTT, and RDMLOG. The format for these directives are found in the following subsections.

5.2.19.1 Identifying alternative resource datasets for RDM

The *RDMRDS Startup directive is used to identify alternative resource datasets for RDM. The default for the resource dataset is the following:

```
DN=$RESDAT,PDN=RESOURCEDATASET,ID=SYSTEM,OWN=SYSTEM
```

Format:

```
*RDMRDS,PDN=pdn[,ID=id][,ED=ed][,OWN=own][,RD=rdpw][,WT=wtpw]
```

PDN=*pdn*	Permanent dataset name of the dataset to be used as the resource dataset. This is a required parameter.
ID=*id*	User identification under which the dataset is saved. Default is no ID used.
ED=*ed*	The edition of the dataset to be accessed. Default is the highest edition.
OWN=*own*	The ownership value associated with the dataset. Default is SYSTEM.
RD=*rdpw*	The read password if the dataset was saved with one. Default is no read password.
WT=*wtpw*	The write password if the dataset was saved with one. Default is no write password.

5.2.19.2 Turning RDM controls on and off

The *RDMSTT Startup directive is used to turn on and off the status of RDM. It is also used to turn on and off charging, security, and disk password checking status, as well as to set the charging rate index value of RDM.

Format:

```
*RDMSTT[RDM=ON ][,CHG=ON ][,SEC=ON ] [CRI=n][,DCO=ON ]
            OFF       OFF       OFF                 OFF

   [,PCO=ON ][,JOB=ON ][,PEO=ON ]
         OFF       OFF       OFF
```

RDM=ON OFF	Turns RDM on or off at STARTUP time
CHG=ON OFF	Turns charging on or off at STARTUP time

SEC=ON
OFF
Turns security on or off at STARTUP time

CRI=*n* The charging rate index at STARTUP time. This parameter selects the rate for job charging.

DCO=ON
OFF
Allows disk space checking to be turned on or off at STARTUP time. This parameter is used to control the user limits for permanent archive datasets.

PCO=ON
OFF
Allows password checking to be turned on or off at STARTUP time. This parameter is used to control the password checking on the account statement.

JOB=ON
OFF
Allows job checking to be turned on or off at STARTUP time. This parameter is used to control the job acceptance.

PEO=ON
OFF
Allows password expiration to be turned on or off at STARTUP time. This parameter is used to control the expiration of user passwords in the Resource Dataset.

5.2.19.3 RDM trace options

The *RDMLOG command is used to set the LOG option for RDM.

```
*RDMLOG, LOG=option
```

LOG=*option*

Selects or changes the Resource Dataset Manager logging options. Any combination of the defined options in the following list or any additional options may be used.

FULL	Turns on all RDM Cray logging options
JOB	Log job control functions information
ERROR	Log error information
DISK	Log disk function information
QUEUE	Log QUEUE function information
TRACE	Log subfunction commands including the RDD
OF	Turns all RDM logging off
SITE	Turns on all RDM site logging options
SITE1	Site 1 defined logging options
SITE2	Site 2 defined logging options
SITE3	Site 3 defined logging options

5.3 THE INSTALLATION-DEFINED ACCOUNTING ALGORITHM

An installation can establish an accounting algorithm that measures resource usage by manipulating the values of weighting factors set by installation parameters.

Several resources are considered in developing an accounting algorithm for an operating system. Each site can have its own criteria in determining what resource usage to include in an accounting algorithm. Installation parameters defined in STP (and described in appendix A, Installation Parameter Listing) determine the relative weight of each resource.

The resources to be included in the accounting algorithm are those whose weighting factors are greater than 0. Each weighting factor is multiplied by its resource usage, and sums of all the results give the total system billing units (SBUs).

A site may want its algorithm to be independent of the job mix conditions, since the same job running at different times could encounter a variety of mix conditions. To ensure such independence, certain data must be omitted from the accounting algorithm. Because physical I/O requests, I/O wait time, time waiting for a JXT, and time waiting to execute vary from run to run, they should not be included in such an accounting algorithm. A site can omit them by setting the weighting factor to 0 for each of the resources.

Accounting information such as the time and date of run, job class, job priority, and termination status are also gathered by the system but are not included in the calculation of total SBUs. They are stored in a binary record in the system log along with the accounting information normally accumulated at job termination. For more information concerning the operation of the accounting mechanism, see the COS Accounting Aids Internal Reference Manual.

Each resource usage parameter is assigned a weighting factor through an installation parameter. Table 5-1 identifies and describes the purpose of each installation parameter. The number of SBUs used by a resource is calculated as follows:

    SBU=Computer Resource Usage * Weighting Factor

Examples:

    Permanent Dataset Space Accessed * Weighting Factor
    PFA=500 blocks, I@PFA=.2 billing units/block
    SBU=500 blocks *.2billing units/block=100 billing units

    Temporary Dataset Space Used * Weighting Factor
    TFS=500 blocks, I@TFS=1 billing unit/block
    SBU=500 blocks * 1 billing unit/block=500 billing units

Disk Sectors Moved * Weighting Factor
IOB=1016 blocks, I@IOB=2 billing units/block
SBU=1016 blocks * 2 billing units/block=2032 billing units

The total number of SBUs:

total system billing units (TSBU)
=(Computer Resource Usage * Weighting Factor)
=[(PFA*I@PFA)+(PFS*I@PFS)+...+(BRF*I@BRF)+(BSF*I@BSF)]

If the TSBU is less than an installation parameter defining a minimum TSBU (I@TSBU), the I@TSBU value is used.

If a weighting factor is set to the default of 0, that resource usage is excluded from system billing. Table 5-1 shows the various accounting parameters and the units in which they are accumulated. Table 5-1 also shows the usual unit that is charged for, and the corresponding weighting factor required to charge one SBU for one unit used.

Four parameters (TSD, TSW, TWJ, and TSX) are accumulated in time-stamp units. To count one SBU for a usage of 1 second of each of these four parameters, the corresponding weighting factor should be $1.024*10^{-9}$. This applies to CRAY X-MP computer systems only; CRAY-1 A, B, M, and S computer systems need no change.

The memory integrals (XMI and DMI) are accumulated in word-seconds. To count one SBU for a resource usage of 1 Mword/second, the corresponding weighting factor should be $1.0*10^{-6}$.

Suitable factors for MIM and MXM, might be $1.0*10^{-3}$ for Kwords or $1.0*10^{-6}$ for Mwords.

The values given in the preceding paragraph, along with the remaining parameters, can be multiplied by a value in the range 0.001 to 5.0 to obtain a suitable formula. The following accounting formula is an example.

$$SBU=TSX + \frac{DMI + XMI}{10} + \frac{IOTIME}{30} + \frac{PFS}{100} + \frac{BRF + 2 * BSF}{100}$$

In this formula, IOTIME is 0.03*IOR + 0.001*IOB (30 ms per seek and 1 ms per block transfer). The weighting factors required in this formula are as follows:

Weighting Factor	Value
I@TSX	$1.024*10^{-9}$
I@DMI	$1.0*10^{-6}$
I@XMI	$1.0*10^{-6}$
I@IOR	$1.0*10^{-3}$

Weighting Factor	Value
I@IOB	$3.3*10^{-5}$
I@PFS	$1.0*10^{-2}$
I@BRF	$1.0*10^{-2}$
I@BSF	$2.0*10^{-2}$

Table 5-1. Assignment of Weighting Factors to System Resources

Parameters	Weighting Factor	Description	Usual Accounting Unit	Required Weighting Factor
PFA	I@PFA	Permanent dataset space accessed (blocks)	Blocks	1.0
PFS	I@PFS	Permanent dataset space saved (blocks)	Blocks	1.0
TFS	I@TFS	Temporary dataset space (blocks)	Blocks	1.0
IOB	I@IOB	Disk sectors moved	Blocks	1.0
IOF	I@IOF	FSS sectors moved	Sectors	1.0
IOR	I@IOR	Physical I/O requests	Requests	1.0
SMI	I@SMI	Memory integral (Wait semaphore time)	Megaword seconds	$1.0*^{-6}$
TSD	I@TSD	I/O wait time (time-stamps)†	Seconds	$1.024*10^{-9}$
TSW	I@TSW	Time waiting to execute (time-stamps)†	Seconds	$1.024*10^{-9}$

† Time-stamps are expressed in units of 1/1.024 nanoseconds; that is, 1 time-stamp represents 0.97656 nanoseconds and 1 nanosecond is 1.024 time-stamps.

Table 5-1. Assignment of Weighting Factors to System Resources (continued)

Parameters	Weighting Factor	Description	Usual Accounting Unit	Required Weighting Factor
TSWS	I@TSWS	Wait semaphore time (time-stamp)	Seconds	$1.024*10^{-9}$
TWJ	I@TWJ	Time waiting for a JXT (time-stamps)†	Seconds	$1.024*10^{-9}$
MRD	I@MRD	Memory-resident datasets		1.0
OPC	I@OPC	OPEN calls		1.0
CLC	I@CLC	CLOSE calls		1.0
MIM	I@MIM	Minimum memory used (words)	Kwords Mwords	$1.0*10^{-3}$ $1.0*10^{-6}$
MXM	I@MXM	Maximum memory used (words)	Kwords Mwords	$1.0*10^{-3}$ $1.0*10^{-6}$
XMI	I@XMI	Memory integral at execution time	Mword-seconds	$1.0*10^{-6}$
DMI	I@DMI	Memory integral (I/O wait time)	Mword-seconds	$1.0*10^{-6}$
TSX	I@TSX	CPU time (time-stamps)†	Seconds	$1.024*10^{-9}$
BRF	I@BRF	Number of sectors received from front-end	Blocks	1.0
BSF	I@BSF	Number of sectors sent to front-end	Blocks	1.0
TBM	I@TBM	Tape blocks moved	Blocks	1.0
TSM	I@TSM	Tape sectors moved	Blocks	1.0

† Time-stamps are expressed in units of 1/1.024 nanoseconds; that is, 1 time-stamp represents 0.97656 nanoseconds and 1 nanosecond is 1.024 time-stamps.

Table 5-1. Assignment of Weighting Factors to System Resources (continued)

Parameters	Weighting Factor	Description	Usual Accounting Unit	Required Weighting Factor
TVM	I@TVM	Tape volumes mounted	Blocks	1.0
GRR*n*†	GRWF*n*	Resource units reserved	Blocks/ tape units	1.0

† GRR is calculated for each generic resource used by the job. The weighting factor GRWF*n* is specified on the GRT entry BUILD directive. The maximum number of generic resources is limited to 16.

5.4 STARTUP AND SHUTDOWN PROCEDURES

Primary reasons for executing a system startup or shutdown are system power-on/power-off, system failure, or changing systems. Whatever the reason, following proper procedures minimizes the chances of losing jobs, time, and information. The I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051, describes the startup procedures for a Cray computer system with an IOS. Subsection 5.5.3, Nonfatal Error Conditions, describes the Startup messages included in the following subsections.

5.4.1 STARTUP OF A CRAY COMPUTER SYSTEM

After the computer system has been powered on and parameter files are prepared, the system is ready for Startup.

5.4.1.1 Startup procedure using a Data General system

Startup using the Data General system is performed as follows. Clear the MCU channel, start the system, and watch for Startup completion by typing the following station commands:

DUMP D 0 1
START *COS-filename parameter-filename*
LOG DG
STM

```
master device name -DEVICE LABEL UPDATE FLAG SET (MASTER)
REPLY 'GO'   TO RE-WRITE DEVICE LABEL
REPLY 'SKIP' TO IGNORE UPDATE FLAG
```

Respond with

REPLY *msg number* GO

5.4.1.2 Startup procedure using an IOS

Startup for a Cray computer system using an IOS is described in detail in the I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051. Use the following procedure if the Kernel is already running:

CTRL-D

The system responds with SYSDUMP?. Reply to the query by entering:

N

The system responds with RESTART?. Reply to the query by entering:

Y

The system responds with FILE @DK0?. Reply to the query by entering:

KERNEL (or whatever file name is used for the desired IOS system)

When the Kernel comes up, enter:

START *COS-filename parameter-filename*
STATION

Enter the following commands from the station console:

LOG
STM

Startup issues an operator message in response to the STM station display, such as:

ENTER CONFIGURATION CHANGES OR CONTINUE

Respond to the message by entering:

REPLY *msg-number* CONTINUE
or
REPLY *msg-number* GO

The system responds with a message (as described for Data General Startup). Reply to the message and then enter:

```
Y.
```

5.4.1.3 Y display

The Y display brings up the Exchange Package for the Startup task by default when using either the Data General or the IOS. When Startup is complete, the following message appears in the Exchange Package to the right of the S1 and S2 registers:

```
STARTUP COMPLETE
```

For all Cray computer systems, submit and run any necessary system jobs and check the output before allowing user jobs to run (this is unnecessary if *SUBMIT directives are used in the parameter file). One recommended system job is an EXTRACT job to get the TYPE=STARTUP message for the Startup just completed. This message tells if Startup has encountered any problems during the execution of its task. JCSDEF or a job to change entries in the SDR can also be run.

When the system jobs have successfully completed, user jobs should be allowed to execute. Resume any recovered jobs with the RECOVER command. The RESUME,ALL command also resumes jobs that were selectively suspended, but no others execute until the LIMIT command is employed. The appropriate value for the LIMIT command depends partly on the job class structure.

Now that jobs already in the CPU are executing, the CHANNEL command can be entered to allow acceptance of more jobs from front-end computers other than the master operator station. Usually, initializing the front-end station and using the LOGON command is necessary at this time. The CHANNEL command is needed only when the LCT entry has been assembled OFF, or if the *LCT directive turned the channel OFF during Startup.

5.4.2 SHUTDOWN OF A CRAY COMPUTER SYSTEM

Before powering down the CRAY Y-MP, CRAY X-MP EA, CRAY X-MP, or CRAY-1 CPU, the system should be idled down so that executing jobs can later resume where they left off. Even if no jobs are executing, these procedures minimize the chances of problems occurring. Jobs that have magnetic tape drives assigned cannot be resumed, but will be restarted from the beginning if possible.

First, prevent additional jobs from entering the computer system by turning off all channels having a front-end logged on using the following command:

```
CHANNEL n OFF
```

If the front-ends are logged on through the IOS, use the following command (where *n* is the number of the MCU channel pair and *m* is the ordinal associated with the IOS pair):

```
CHANNEL n,m OFF
```

Do not attempt to turn off the channel/ordinal of the master operator station.

The SHUTDOWN command idles all jobs and prepares them for later recovery. When the status display shows that all jobs are in an idle state, type the following command to cause the System Performance Monitor (SPM) task to record the statistics for the last interval.

```
INITIATE 8
```

Next, an EXTRACT job is run to flush the system log buffers.

5.4.2.1 Shutdown using a Data General system as MCU

The CPU is stopped for a Cray computer system using a Data General system as the MCU by entering the following command:

```
DUMP D 0 1
```

Disks can now be put off-line and the system can be powered down.

5.4.2.2 Shutdown for a Cray computer system with an IOS

The following commands can be issued from the IOS at the Kernel console:

```
CTRL-D
```

The system responds with SYSDUMP?. Reply to the query with:

```
N
```

The system responds with RESTART?. Reply to the query with:

```
N
```

The disks can now be put off-line and the system can be powered down.

5.4.3 CHANGING SYSTEMS

Special precautions must be taken when changing systems. Not taking such precautions can lead to the loss of system log information, failure of jobs to recover, or even failure of the system to start. These precautions are as follows:

- If any differences exist between the structures or sizes of certain significant tables contained in the Job Table Area (JTA) for a job, jobs that were executing cannot be recovered. For this reason, jobs should not be running if the present system will be replaced by an incompatible system. If such a change is necessary, a restart with *RRJ,0 forces all jobs to be rerun. If these tables change size, Startup halts unless you specify *RRJ,0.

- If the system memory size as defined in the installation parameter C@MMSIZE (see appendix A) is different for the two systems, the system log buffers cannot be recovered. In this case, be sure to run EXTRACT to flush the log, and then stop the CPU by using 200=1,E.

- If the master device is different for the two systems (that is, if two independent sets of disks are being used), two precautions must be taken:

  1. You must run EXTRACT to flush the log buffer and then stop the CPU.

  2. A field engineer must clear memory before the other system is started, a *SYSLOG parameter file entry may be used to force reinitialization of the log, or a *SUPSYS can prevent the recovery of the buffer and DSP.

Without these precautions, the Dataset Parameter Table for the system log is still recovered, still contains the position of the first system log, and begins writing to the new system log at the same position. By clearing memory, the Log Manager is forced to read the system log and determine its position.

5.4.4 FORCING A DUMP

Generally, a system dump is executed after a system failure, and the dump is later analyzed to determine the cause of the problem. Situations arise, however, where a problem shows up in the system, but the system does not crash. Dumping a system in this state is sometimes helpful.

Before performing a dump, determine whether the SHUTDOWN command will significantly change the state of the system, increasing the difficulty of finding the cause of the problem. The advantage of a shutdown is that all jobs, except those using magnetic tapes, will recover after the dump and restart. The disadvantage of a shutdown is that jobs may not be caught in their executing state.

5.4.4.1 Forcing a dump from the Data General system

For a shutdown of a Data General system MCU without an IOS, enter the following sequence of commands:

SHUTDOWN

(wait for all jobs to be suspended)

SYSDUMP

If you do not want a shutdown, enter the following command from a Data General MCU:

SYSDUMP

5.4.4.2 Forcing a dump from the IOS

For a shutdown, enter the following sequence of commands:

SHUTDOWN (from the station console only)

(wait for all jobs to be suspended)

CTRL-D (from the MIOP Kernel console only)

If you do not want a shutdown, enter the following command at the MIOP Kernel console:

CTRL-D

The system responds with SYSDUMP?. Reply to the query by entering:

Y

5.4.5 WHEN COS DOES NOT RESPOND TO LOGON

Failure of COS to respond to the LOGON station command after a restart could mean a serious problem exists. The way the situation is handled determines the status of jobs in the system when it went down, whether a new system log needs to be created, and if an install is required. Some common problems can be resolved with one of the procedures described next.

First, check if a hardware failure, such as a disk fault, has occurred. If the problem is a hardware fault, the field engineers must resolve it.

Next try restart again (see the I/O Subsystem (IOS) Operator's Guide for COS for IOS restart procedures) after clearing the MCU channel with the following command:

```
DUMP D 0 1  (for a Data General MCU without an IOS)
```

or you can issue the following command at the MIOP Kernel console:

```
CTRL-D
```

If this does not work, a dump should be taken through the station performing maintenance control functions. Do not use the SYSDUMP command; if the system does not start, the dump cannot be retrieved. The areas to dump are as follows:

- EXEC history trace
- STP tables
- All of STARTUP subroutine Z

These should be analyzed, beginning with the hang message (if it exists) in STP. Next analyze the trace, and then the argument values stored in Z.

If the problem has not been determined and is not resolved, use any of the following procedures:

- Have the field engineers run CPU and disk diagnostics
- Try a restart with *RRJ,0; that is, do not try to recover rolled jobs.
- Try a deadstart
- Force a new edition of the system log by adding a *SYSLOG directive to the parameter file
- Breakpoint through Startup to force it around a problem

If the system still does not start, gather as much information as possible. Do not do an install until the analyst-in-charge (AIC) of the system and the site operations manager approve it.

If the install does not work either, a hardware problem almost certainly exists.

5.4.6 PARAMETER FILE ERROR PROCESSING

During processing of the parameter file, the ZY routine maintains an internal buffer, which is used to hold messages resulting from error conditions found on parameter file directives. This buffer is dynamically allocated, and resides above the highest addresses occupied by the COS binary and CSP. The buffer address can be determined through an indirect reference through the task-pointer-area in STPTAB. One word in that area contains the address of the buffer. Normally it is unnecessary to determine the location of the buffer. When ZY terminates, the main STARTUP routine Z begins execution, and displays any messages that ZY has constructed at the master operator console. A response is required. The message format is:

```
PARAMETER FILE DIRECTIVE CONTAINS ERROR - DIRECTIVE IGNORED.
THE DIRECTIVE IN ERROR IS DISPLAYED BELOW.
REPLY 'GO' TO CONTINUE STARTUP PROCESSING.
```

text of invalid directive

Any reply other than GO causes the message to be redisplayed with the additional line:

text   INVALID. CORRECT AND RE-ENTER.

5.5 STARTUP ERROR RECOVERY

When System Startup encounters an error, the operator station console displays an appropriate message. Because this message generally requires operator interaction, all Startup operator messages and explanations are placed together in this subsection.

5.5.1 FATAL ERROR CONDITIONS

All fatal error conditions cause Startup to issue an operator message and loop rather than crashing. Error messages arising from a DQM or PDM error return status to Startup. The error message includes error status. Some messages begin with STARTUP CANNOT CONTINUE. This portion of the message has been omitted in the alphabetical list of fatal error messages.

Message	Meaning	Action
CANNOT SET UP UNIQUE ACCESS FOR THE DXT-PDS IS FULL	No space remains in the PDS to build an entry for the Dataset Catalog Extension.	Check size of the PDS and determine if it has changed since previous system.

Message	Meaning	Action
DISK SPACE NEGATIVE WHILE RECOVERING ROLLED JOBS	Amount of available space on a disk went negative while allocating space for rolled jobs.	Fatal, internal error. Consult the CRI site analyst.
DQM ERROR ON REWRITE OF JTA DURING RRJ	DQM returned an error status after unsuccessfully attempting to write the JTA of a recovered rolled job.	Fatal; check DQM error status; might be able to circumvent problem by not recovering rolled jobs.
DQM ERROR DEALLOCATING SDT DATASET	DQM encountered an error while trying to delete a spooled dataset from the SDT	Check DQM status; circumvent by doing Deadstart rather than a Restart.
DQM ERROR WHILE ALLOCATING ROLL INDEX	While attempting to write the index buffer to $ROLL, DQM returned an error after being unable to allocate space on the $ROLL file.	Fatal; check DQM error status and ensure adequate space exists on system.
DSC DAT BAD OR FLAW EXISTS IN DSC	While trying to reserve the space for the DSC, either cross allocation has occurred or a bad DAT existed.	Fatal; check if any flaws are in the DSC area or determine if DSC DAT is in error.
$DSC-EXTENSION CAN ONLY BE CREATED DURING AN INSTALL OR DEADSTART	A RESTART was being performed but the $DSC-EXTENSION did not already exist.	Perform either an INSTALL or a DEADSTART.
$DSC EXTENSION CAN'T BE ACCESSED. PDM ERROR=*nnnn*B†	PDM could not access the $DSC-extension dataset.	Fatal; check PDM return status.
$DSC EXTENSION CAN'T BE ADJUSTED-PDM ERROR=*nnnnB*†	PDM could not increase the size of the $DSC-extension dataset.	Fatal; check PDM return status.

† The PDM error return code can be 3 or 4 digits in length.

Message	Meaning	Action
$DSC EXTENSION IS FULL. ANOTHER STARTUP MUST BE PERFORMED WITH A REQUEST TO INCREASE THE SIZE OF THE DXT.	Every entry in the $DSC-extension dataset is in use.	Increase the size of the DXT by an INSTALL, DEADSTART, or WARMSTART.
DXT COULD NOT BE SAVED. PDM ERROR	PDM could not save the $DSC-extension dataset.	Fatal; check the return status.
ENTRIES CONTAINED IN RECOVERED $SDR EXCEED AVAILABLE SPACE	Recovered System Directory Dataset contains more entries than can be recovered in the current system's SDR.	Use *SDR directive to circumvent problem.
INTERNAL ERROR - BAD DUMP DAT	When moving the system dump DAT to the first sector of the dataset, errors were detected in the DAT.	Fatal; check system dump DAT for errors.
INTERNAL ERROR - BAD ZSTOP ERROR CODE	The fatal error routine entered with an unknown error message number. Startup problem.	Fatal, internal error. Consult CRI site analyst.
INTERNAL STARTUP/ CONFIGURATION ERROR	Startup encountered an error condition that precludes continuation. May be caused by configuration errors or serious software/hardware problems.	Fatal; contact a system analyst.
I/O ERROR ENCOUNTERED READING THE $DSC-EXTENSION	DQM returned an error status during I/O operation on the DXT.	Check the error status in the DXT DSP.
I/O ERROR ENCOUNTERED WRITING THE $DSC-EXTENSION	DQM returned an error status during I/O operation on the DXT.	Check the error status in the DXT DSP.
I/O ERROR OCCURRED INITIALIZING THE DXT ON DISK	DQM returned an error status during I/O operation on the DXT.	Check the error status in the DXT DSP.

Message	Meaning	Action
I/O ERROR READING *JCLASS DATASET	DQM encountered an error while reading the dataset specified in a *JCLASS directive.	Check DQM status. Circumvent by removing *JCLASS directive.
I/O ERROR READING JOBCLASSROLLED DATASET	DQM encountered an error while trying to read the Job Class Structure dataset.	Fatal; check DQM error status.
I/O ERROR WHILE WRITING SYSTEM DIRECTORY	DQM encountered an error writing the System Directory Dataset.	Fatal; check DQM error status.
I/O ERROR WRITING JOB CLASS STRUCTURE DATASET	DQM encountered an error while trying to write the Job Class Structure dataset.	Fatal; check DQM error status.
I/O ERROR RETURN FROM DQM WHILE EXTENDING THE DXT	DQM returned an error status during an I/O operation on the DXT.	Check the error status in the DXT DSP.
I/O ERROR WHILE READING THE SYSTEM DIRECTORY	DQM returned an error status after an unsuccessful read of $SDR.	Fatal; check DQM error status. An *SDR in parameter file may circumvent problem.
I/O ERROR WHILE WRITING ROLL INDEX DATASET	DQM returned an error status after unsuccessful write of index buffer to the $ROLL dataset.	Fatal; check DQM error status.
INSUFFICIENT MEMORY FOR STARTUP TO EXECUTE	Startup and its associated tables exceeded available field length for system tasks.	Fatal; determine reason for change in Startup size.
MEMORY POOL SPACE INSUFFICIENT FOR TEXT	When recovering I/O datasets, space in the text memory pool could not hold all of the text.	Fatal; check if memory pool, I@MP1SZ, has changed since last running system. May require a deadstart.

Message	Meaning	Action
MULTIPLE MASTER DEVICES	More than one device has been defined as the master device; may be caused by information in the label or EQT.	Operator option to select the correct master device.
NO MASTER DEVICE FOUND DURING DEADSTART/RESTART	No device contains the master device flag in the label.	Fatal; a master device has to exist because a DSC has to be pre-defined during a deadstart or restart.
NO ROOM IN STP DAT FOR DSC DATASET	STP tables DAT area is not large enough to contain the DAT for the DSC.	Fatal; check SZ@DAT and the size of the DSC DAT to ensure that space exists.
NO ROOM IN STP DAT TO RECOVER PERMANENT DATASET	During permanent dataset recovery, insufficient space in the STP DAT area for moving the DAT for a dataset to memory from the DSC.	Fatal; determine if STP DAT size has decreased since the previously running system.
NO SPACE REMAINS IN AUT FOR RECOV-ERED INTERACTIVE INPUT DATASET	Startup found more interactive jobs to recover than there was space in the current system's AUT table.	Circumvent by doing a Deadstart rather than a Restart.
PDM ERROR DELET-ING SDT DATASET	PDM encountered an error while deleting a spooled dataset from the SDT.	Check PDM error status. May be circumvented by doing a Deadstart
PDM ERROR ON ADJUST OF JOB-CLASSROLLED DATASET	PDM returned an error while trying to adjust the size of the JOBCLASSROLLED dataset.	Fatal; check PDM error status.
PDM ERROR ON SAVE OF JOB CLASS STRUCTURE DATASET	PDM returned an error while trying to save the job class dataset.	Fatal; check PDM error status.
PDS IS FULL	No space in PDS to build an entry for an accessed permanent dataset. This dataset could be a system dataset, a dataset in the SDR, or a permanent dataset accessed by a recovered rolled job.	Fatal; check size of the PDS and determine if it has changed since previous system.

Message	Meaning	Action
STARTUP ABORT REQUESTED BY OPERATOR	Operator chose to stop Startup after an error occurred while processing a *SUBMIT directive.	Fix or remove the *SUBMIT directive.
SYSTEM DIRECTORY DATASET IS BAD	Startup determined that the $SDR dataset has been destroyed.	Fatal; check error, may have to include *SDR in parameter file.
TABLE SIZE CHANGED - CAN'T RECOVER ROLLED JOBS	System changed since the last running system and the length of the JXT, DNT, or Rolled Job Index has changed.	Operator has option to continue without recovery of rolled jobs.
TWO DEVICES OF CONFLICTING TYPE CONFIGURED ON SAME PATH	Multiple logical devices on the same physical device must have matching device types.	Use DT parameter on CONFIG command or *CONFIG parameter directive to correct device type.
TWO LOGICAL DEVICES HAVE OVERTRACK RANGES	Two or more logical devices configured on the same physical device overlap because of bad EQT parameters STK and NTK.	Use STK and NTK parameters on CONFIG command or *CONFIG parameter directive to correct overlap.
UNABLE TO ALLOCATE DISK SPACE FOR FLUSH DATASET	DQM encountered an error trying to allocate space for a volatile device backup dataset.	Check DQM error status. Circumvent by making device nonvolatile or by downing the device.
UNABLE TO ALLOCATE THE SYSTEM DUMP AREA	Space for the system dump preallocated area could not be obtained. Either no space remained or allocated space could not be read.	Fatal; space must be allocated on the master device. Check if space exists or determine I/O error.
UNABLE TO CREATE DSC DATASET	During creation of the DSC as a blocked dataset, Startup could not write one of the pages.	Fatal; check DQM error status.

Message	Meaning	Action
UNABLE TO WRITE CSP TO DISK	DQM returned error status on all attempts to write CSP to disk.	Fatal. For all I@NCSP copies of CSP, DQM returned errors; check DQM error status.
UNABLE TO WRITE DAT TO SYSTEM DUMP	DAT for the system dump area could not be written to the first sector of this area.	Fatal; check DAT for error or determine if I/O error occurred.
UNEXPECTED PDM ERROR ON ACCESS OF $SDR	PDM returned an error while trying to access the System Directory Dataset.	Check PDM error status. Circumvent by using *SDR parameter file directive.
UNEXPECTED PDM ERROR ON RELEASE OF ROLL INDEX DATASET	I/O error occurred on the read of $ROLL; Startup then tried to release the bad dataset and PDM returned an error status.	Fatal; check PDM and DQM error statuses.
UNEXPECTED PDM ERROR ON SAVE OF ROLL INDEX DATASET	PDM issued a bad response to the Startup request sent to save the Rolled Job Index.	Fatal; check PDM error status to determine error type.
UNEXPECTED PDM ERROR ON SAVE OF SYSTEM DIRECTORY DATA-SET	PDM returned an error while trying to save $SDR.	Fatal; check PDM error status.
UNEXPECTED PDM STATUS ACCESSING FLUSH DATASET	PDM returned an error while trying to access a previously created volatile device backup dataset.	Check PDM error status. Circumvent by making device nonvolatile, by downing the device, or by removing the *RESTORE directive.
UNEXPECTED PDM STATUS SAVING FLUSH DATASET	PDM returned an error while trying to save a newly allocated volatile device backup dataset.	Check PDM error Status. Circumvent by making device nonvolative or by downing the device.

5.5.2 RDM FATAL ERRORS

RDM issues the following fatal messages when problems are detected during startup.:

RDM non FATAL ERROR TYPE- *type* STATUS *status*
FUNCTION *fn name* RD ENTRY - *rd entry key* ADDRESS *rdm addr*

RD001 - RDM UNABLE TO ACCESS RESOURCE DATASET
Occurs when PDM cannot access Resource Dataset (RD). Also, the RD could have incorrect identifier, owner, edition, or permissions. Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Causes RDM to be turned off and the system comes up without RDM active
GO	Causes RDM to rewrite the system date in the RD and continues operation

RD003 - UNRECOVERED DISK ERROR ON RD (READ) - SEE ERROR
Occurs when there is a disk error during a read operation for the RD information entry. Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Causes RDM to be turned off and the system comes up without RDM active
GO	Same as ABORT

RD004 - UNRECOVERED DISK ERROR ON RD (WRITE) - SEE ERROR
Occurs when there is a disk error during a write operation for the RD information entry. Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Cannot be used
GO	Causes RDM to skip the rewrite of the RD information entry and continues operation

RD030 - RDM INFORMATION ENTRY NOT FOUND - GO OR ABORT.
Caused when RD is corrupted or the dataset specified is not a RD. Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Causes RDM to be turned off and the system comes up without RDM active
GO	Same as ABORT

RD031 - RDM INFORMATION ENTRY IS CORRUPT - GO OR ABORT.
Caused when RD information entry is corrupt due to a checksum error. Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Causes RDM to be turned off and the system comes up without RDM active
GO	Same as ABORT

RD032 - RDM INFORMATION ENTRY I/O ERROR - CANNOT UPDATE.
Caused by disk error occurring during a write operation for the RD information entry. Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Causes RDM to be turned off and the system comes up without RDM active
GO	Causes RDM to skip the rewrite of the RD information entry and continues operation

RD033 - RDM INFORMATION ENTRY FLD RD*xxxx* - BAD VALUE.
Caused when RDM detects a bad value in the *xxxx* field in the RD information entry. The following is a list of the values and their descriptions:

RIDKF	Job decay factor
RIFAV	Filespace average constant
RIQK1	Queue calculation constant 1
RIQK2	Queue calculation constant 2
RIQK3	Queue calculation constant 3
RIQAR	Queue aging shift constant

Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Causes RDM to be turned off and the system comes up without RDM active
GO	Same as ABORT

RD034 - RDM INFORMATION ENTRY FLD RIRDNP - SIZE MIS-MATCH

Caused when RDM detects an incorrect size for the dataset in the RD information entry. Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Causes RDM to be turned off and the system comes up without RDM active
GO	Same as ABORT

RD035 - DATE IS EARLIER THAN LAST RESTART - INVESTIGATE.

Caused when RDM detects a mismatch between the current system startup time and the time fields in the RD information entry. Allowable operator responses are as follows:

CRASH	Crashes the system and allows a system dump to be taken
ABORT	Causes RDM to be turned off and the system comes up without RDM active
GO	Same as ABORT

5.5.3 NONFATAL ERROR CONDITIONS

During Startup, various error conditions are reported to the operator. The operator can try to correct the error causing condition while continuing the startup process. All messages sent by Startup to the ZYLOG buffer are also displayed for immediate operator interaction. The message and parameter file command being processed is displayed. An operator response is required before Startup continues.

5.5.3.1 Altering the Configuration Table and Equipment Table

If an IOS is configured in the system, Startup attempts to initialize a communication link with the Kernel in the Master I/O Processor (MIOP). Startup makes five attempts, at 4-second intervals, to initialize the link. If the link cannot be successfully initialized, Startup attempts to send the following message to the system operator:

```
UNABLE TO COMMUNICATE WITH I/O SUBSYSTEM
ENTER GO TO SKIP CONFIGURATION CHANGES
ANY OTHER REPLY CAUSES RETRY
```

Normally, in a system with the IOS configured, the system operator is at the IOS console. It is possible, however, for the operator to be at any station. A reply of GO causes Startup to stop attempting to initialize the link with IOS, and also causes Startup not to attempt to get configuration changes. Any other reply causes Startup to make five more attempts to initialize the link.

Once the link has been established, or if there is no IOS configured, Startup will ask the master operator to enter any desired configuration changes by issuing the following message:

```
ENTER CONFIGURATION CHANGES OR CONTINUE
```

To alter the device configuration (add or delete a device or change the status of a device listed in the Configuration Table (CNT) or Equipment Table (EQT)), enter the following command at the master operator station:

REP *msgnum* CONFIG *parameters*

The parameters to be entered are those given with the CONFIG command in the I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051, or the Data General Station (DGS) Operator's Guide, publication SG-0006.

A maximum of 72 characters can be entered at a time. If more than 72 characters are needed, multiple replies can apply to the same device. The requested change in the configuration occurs if Startup does not detect any errors in the reply. If errors are detected, Startup issues another message:

INVALID ENTRY. CORRECT AND REENTER

When the configuration changes have been reentered successfully, the operator continues Startup by entering the following:

REP *msgnum* CONTINUE or REP *msgnum* GO

Startup now verifies the CNT entries to ensure that the operator's changes are consistent with the device types.

Inconsistencies generate the following message:

CNT ENTRY INVALID - DEVICE = *devname* ENTER ANY REPLY TO CONTINUE

Enter any reply. Startup does not check for acceptable characters.

Startup then reissues the initial request for configuration changes and cannot be completed until all errors in the CNT are corrected through valid operator response.

5.5.3.2 Checking the Engineering Flaw Table

Following the configuration change processing, Startup searches each configured DD-19, DD-29, DD-39, DD-40, or DD-49 Disk Storage Unit (DSU), SSD, and Buffer Memory for an Engineering Flaw Table (EFT). If a device is found without an EFT, a separate message for each such device is issued. (The message is not issued for device types SSD or EBM, since it is not considered an error for EFTs to be missing on these devices.) The following message is among those displayed when STM is entered at the master operator station:

```
MN  TIME  devname   NO ENGINEERING FLAW TABLE FOUND
           ENTER GO TO CONTINUE STARTUP
             SKIP TO CONTINUE W/O FURTHER WARNINGS
```

If the EFT information is required by the site, the operator notifies the on-site engineers to recreate the flaw table before completing the startup procedure.

To resume startup with further notification of missing EFTs, enter the following:

REP *msgnum* GO

To complete startup with no further notification of missing EFTs, enter the following:

REP *msgnum* SKIP

To disable the response-required EFT message, use the *SKIPEFT parameter file directive described in subsection 5.2.7, Disk Flaw Directives.

5.5.3.3 Parameter file processing

When an error is encountered in a parameter file command during parameter file processing, the display message contains the error type that was detected and the parameter file command that was in error. If GO is typed in response to the error message, Startup ignores the parameter file command that was in error. The message displayed is as follows:

```
PARAMETER FILE DIRECTIVE CONTAINS ERROR - DIRECTIVE IGNORED.
THE DIRECTIVE IN ERROR IS DISPLAYED BELOW.
REPLY 'GO' TO CONTINUE STARTUP PROCESSING.
```

5.5.3.4 Label processing

Following parameter file processing, Startup issues the informative message:

STARTUP IS PERFORMING *type*

Type is AN INSTALL, A DEADSTART, or A RESTART. The type of Startup being performed depends on installation options and parameter file contents. After issuing this message, Startup attempts to locate device labels for each device in the EQT that is not marked as not available. If an INSTALL is being performed, it is not considered an error if no label is found on a device. It is an error if no label is found on a device during RESTART or DEADSTART. If a label is found during an INSTALL, the flaw information in the existing label is included in the DRT and in the new label, if one needs to be written later. Flaw information from labels found during DEADSTART/RESTART is also retained.

When Startup is unable to read/write a device label, the operator is given options to down the device or force Startup to try and read/write the device label again. The following messages are output:

```
device UNABLE TO READ DEVICE LABEL ON num TRACKS
       I/O ERROR STATUS WAS RETURNED ON errcnt OF THE TRACKS
REPLY 'GO' TO CONTINUE WITH DEVICE DOWN
REPLY 'RETRY' TO TRY READING LABEL AGAIN
REPLY 'LABEL' TO LABEL DEVICE AND CONTINUE
```

No valid device label was found on the device named. The operator should check to see that the device is on-line and ready. If there is no equipment fault and the label cannot be found, either the device must be configured DOWN or a label must be written. A reply other than GO, RETRY or LABEL causes the message to be displayed again with the additional line:

```
text INVALID.  CORRECT AND RE-ENTER.
```

```
device UNABLE TO WRITE DEVICE LABEL ON num TRACKS
       I/O ERROR STATUS WAS RETURNED ON errcnt OF THE TRACKS
REPLY 'GO' TO CONTINUE WITH DEVICE DOWN
REPLY 'RETRY' TO TRY WRITING LABEL AGAIN
(MASTER DEVICE MAY NOT BE DOWNED)
```

The Startup task was unable to write a device label on the named device. Check to see that the device is on-line and ready. If the condition persists and there is no equipment fault, the device cannot be used and you should notify the site engineers. A reply other than GO or RETRY causes the message to be displayed again with the following additional line:

```
text INVALID.  CORRECT AND RE-ENTER.
```

If the number of tracks and the count of errors received are equal, the problem may be hardware related. Check to see if the device is off-line or if other similar problems exist. When the count of errors received is less than the number of tracks, a bad track probably exists. The solution is to include a *CONFIG,WDL=Y for the device in the parameter file. If the device label that cannot be read is on the master device, the Startup processing is not able to continue.

Device track boundary values in labels that do not match EQT values could cause possible loss of datasets and other labels as follows:

```
DEVICE NAME = device
EQSTK = nnnnn          EQNTK = nnnnn
DVSTK = nnnnn          DVNTK = nnnnn

REPLY 'GO' IF YOU WISH TO CONTINUE STARTUP
```

Startup compared the logical device track boundaries as saved in the device label (DVL) with those in the device's EQT entry and found that they did not match. If there are datasets on the device, they could be lost or recovered incorrectly if the device's logical boundaries have changed. Also, Startup may be unable to find the labels for other logical devices defined on the same physical device if the boundaries have been altered. If the change of logical device boundaries is intentional, and if it is known that no permanent datasets reside on the physical device, then reply GO to redefine the boundaries and continue with startup. A reply other than GO causes the message to be displayed again with the following additional lines:

text INVALID. CORRECT AND RE-ENTER.

device DEVICE LABEL UPDATE FLAG SET *master*
REPLY 'GO' TO RE-WRITE LABEL
REPLY 'SKIP' TO IGNORE UPDATE FLAG

The Startup task is preparing to rewrite the device label for the named device. This is usually due to a change in the flaws specified through parameter file *FLAW or *DELFLAW directives. In some cases, it may be due to allocation of datasets for use by the IOS, or by reallocation of existing system datasets whose allocation information is contained in the device label (system dump area, DSC). If the device is the master device, the *master* indicator above is '(MASTER)'. If it is not the master device, the indicator contains blanks. If the reply is not GO or SKIP, the message is displayed again with the following additional line:

text INVALID. CORRECT AND RE-ENTER.

Once Startup has read all of the device labels, it searches for the master device. When Startup encounters more than one device in the configuration defined as a master device, the following message is displayed:

MULTIPLE MASTER DEVICES DEFINED. ENTER
DEVICE NAME TO BE USED AS A MASTER (IF
THIS IS DEADSTART OR RESTART, THE
SPECIFIED DEVICE MUST HAVE A MASTER LABEL).

To continue Startup, enter the logical device name of the device to be used as the master. If the device specified does not exist in the EQT, the following message is displayed:

SPECIFIED DEVICE IS NOT IN THE EQT.
ENTER A CORRECT DEVICE NAME.

If it is a deadstart/restart and the specified device is not a master device and has no DSC associated with it, the following message is displayed:

```
SPECIFIED DEVICE IS NOT A MASTER DEVICE.  ENTER A CORRECT DEVICE NAME.
```

5.5.3.5 Dataset Catalog processing

Certain permanent datasets may not be recoverable due to errors. These errors fall into the following categories:

- Crossed allocation
- Residence on downed device
- Multitype validation errors
- Catastrophic errors
- Unrecoverable errors while reading/writing the DSC
- Dataset exceeds logical device boundaries

These irrecoverable datasets are either deleted or retained by the system according to operator response. If any dataset is found in error or if the datasets are to be deleted, the following message is displayed:

xx DATASETS WERE MARKED IN ERROR
DUE TO *error type*
OF THESE *num* WERE DELETED
TO CONTINUE STARTUP PROCESSING
TYPE GO

xx	The number of irrecoverable datasets
error type	One of the six previously mentioned errors

When Startup begins processing the DSC to recover permanent datasets, the following message is displayed:

```
THE DATASET CATALOG IS BEING RECOVERED
```

This message is informational and does not require any response. In certain cases, when certain types of error conditions are found that require a second scan of the DSC, the following informational message is issued at the beginning of the second scan:

```
DATASET CATALOG RECOVERY, PASS TWO
```

This message is always accompanied by other error messages indicating the error conditions found. These may include the following response messages:

```
***** ALLOCATION CONFLICT ON PERMANENT DATASET *****
PDN = pdn     ID = id     OWN = owner     ED = ed
DEVICE devname ALLOCATION UNIT alloc
ENTER 'GO' TO CONTINUE WITH THIS DATASET
ENTER 'SKIP' TO CONTINUE THIS DATASET WITHOUT FURTHER WARNINGS
ENTER 'DELETE' TO DELETE THIS DATASET
EITHER 'SKIP,ALL' OR 'DELETE,ALL' CAUSES FURTHER WARNINGS
TO BE BYPASSED FOR ALL SUBSEQUENT ERRORS OF THIS TYPE.
```

Startup detected that an allocation index, specified within the DAT for a permanent dataset, references disk space that is already allocated. This is normally the addition of a flaw for a device. Other causes could be software or hardware failure at some previous time. A reply of GO causes Startup to continue scanning the DAT for this dataset and reporting each occurrence of a conflict. A reply of SKIP causes Startup to stop reporting occurrences of conflicts for the current dataset, although they continue to be reported to the system log. A reply of DELETE causes Startup to delete the dataset. The ALL parameter may be included with either SKIP or DELETE, and causes Startup to assume the operator response to all subsequent messages for all datasets is the same. Any other reply causes the message to be reissued with the additional line:

text INVALID. CORRECT AND RE-ENTER

The following message appears on the master operator console only from Startup. Class, warning.

```
***** CATASTROPHIC ERROR ON PERMANENT DATASET *****
PDN = pdn     ID = id     OWN = owner     ED = ed
ENTER 'GO' TO FLAG DSC ENTRY AND RETAIN DATASET
ENTER 'DELETE' TO DELETE THIS DATASET
EITHER 'GO,ALL' OR 'DELETE,ALL' CAUSES FURTHER WARNINGS
TO BE BYPASSED FOR SUBSEQUENT ERRORS OF THIS TYPE
```

Startup detected that the DSC entry for the dataset contains an error, and the dataset cannot be successfully recovered. Each occurrence of the condition is reported unless the ALL parameter is included in the reply. To retain the dataset in the DSC, reply GO. To delete the dataset, reply DELETE. Any other reply causes the message to be reissued with the following additional line:

text INVALID. CORRECT AND RE-ENTER

The following message appears on the master operator console only. From Startup. Class, warning.

```
***** INCONSISTENT ALLOCATION ON MULTI-TYPE DATASET *****
PDN = pdn     ID = id     OWN = owner     ED = ed
ENTER 'GO' TO FLAG DSC ENTRY AND RETAIN DATASET
ENTER 'DELETE' TO DELETE THIS DATASET
EITHER 'GO,ALL' OR 'DELETE,ALL' CAUSES FURTHER WARNINGS
TO BE BYPASSED FOR SUBSEQUENT ERRORS OF THIS TYPE
```

Startup detected that two or more datasets having the same QDT ordinal have nonidentical DATs. This message occurs once for each subsequent DSC entry having the same QDT ordinal. In pass 2 over the DSC, it occurs for each DSC entry with the QDT ordinal that has not already been flagged. To prevent the operator from having to reply to each message, the ALL parameter may be used. In this case, the operator is not warned of any subsequent inconsistent allocations, even if they belong to a different QDT ordinal set. The errors continue to be reported to the system log. Any reply other than GO or DELETE causes the message to be reissued with the following additional line:

text INVALID. CORRECT AND RE-ENTER.

The following message appears on the master operator console only. From Startup. Class, warning.

```
***** MULTI-TYPE DATASET QDT ORDINAL INVALID *****
ORDINAL IN DSC ENTRY = dsc-ord MAXIMUM VALID ORDINAL = max-ord
PDN = pdn     ID = id     OWN = owner     ED = ed
ENTER 'GO' TO FLAG DSC ENTRY AND RETAIN DATASET
ENTER 'DELETE' TO DELETE THIS DATASET
EITHER 'GO,ALL' OR 'DELETE,ALL' CAUSES FURTHER WARNINGS
TO BE BYPASSED FOR ALL SUBSEQUENT ERRORS OF THIS TYPE
```

Startup has detected that a multitype dataset has a QDT ordinal greater than the maximum allowed in the version of COS being started. This may occur if the installation parameter NE@SDT is changed. To retain the dataset in the DSC and allocate its disk space, reply GO. To delete the dataset, reply DELETE. Each DSC entry containing this error condition causes an error message to be issued unless the ALL parameter is included in the reply, in which case Startup acts as if the operator has replied to each subsequent message with the same reply (for the same error condition). Any other reply causes the message to be reissued with the following additional line:

text INVALID. CORRECT AND RE-ENTER.

The following message appears on the master operator console only. From Startup. Class, warning.

```
***** PERMANENT DATASET RESIDES ON DOWN DEVICE *****
DEVICE = device
PDN = pdn     ID = id     OWN = owner     ED = ed
ENTER 'GO' TO FLAG DSC ENTRY AND RETAIN DATASET
ENTER 'DELETE' TO DELETE THIS DATASET
EITHER 'GO,ALL' OR 'DELETE,ALL' CAUSES FURTHER WARNINGS
TO BE BYPASSES FOR SUBSEQUENT ERRORS OF THIS TYPE
```

Startup detected that the DAT for the named permanent dataset references one or more devices whose EQT entry is either not present or marked unavailable. Each occurrence of the condition is reported unless the ALL parameter is included in the reply. To retain the dataset in the DSC and to allocate any space that may be on available devices, reply GO. To delete the dataset, reply DELETE. Any other reply causes the message to be reissued, with the additional line:

text INVALID. CORRECT AND RE-ENTER.

From Startup. Class, warning.

```
***** DATASET EXTENDS BEYOND LOGICAL DEVICE BOUNDARIES *****
DEVICE = device
PDN = pdn    ID = id    OWNER = owner    ED = ed
ENTER 'DELETE' TO DELETE THIS DATASET
ENTER 'DELETE,ALL' CAUSES FURTHER WARNINGS TO
BE BYPASSED FOR ALL SUBSEQUENT DELETIONS OF
DATASETS OVERFLOWING DEVICE BOUNDARIES
```

Startup detected that a dataset no longer lies within the bounds of the logical device. This happens if the STK or NTK parameters on the EQT entry for the device have been changed since the creation of the dataset, causing the dataset to no longer lie within the device boundaries. To delete the dataset, type DELETE. Each DSC entry containing this error will cause an error message to be issued unless the ALL parameter is included in the reply. Any other reply causes the message to be reissued with the additional line:

text INVALID. CORRECT REPLY AND RE-ENTER

The following message appears on the master operator console only. From Startup. Class, warning.

If Startup detects block number errors while reading the DSC, or if DQM detects errors while performing I/O on the DSC, Startup issues one or more of the following messages and waits for operator response. Each message indicates appropriate responses. See the COS Message Manual, publication SR-0039, for more detailed explanations of causes and results from the various replies.

```
***** BLOCK NUMBER ERROR IN DSC *****
BLOCK EXPECTED = blk   BLOCK FOUND = blk
REPLY 'GO' TO CLEAR PAGE - DATASETS MAY BE LOST
REPLY 'RETRY' TO RETRY OPERATION
```

```
*** DQM ERROR errcode WHILE operation DSC ***
REPLY 'GO' TO ATTEMPT RE-ALLOCATION OF TRACK
REPLY 'RETRY' TO RETRY LAST REQUEST
*** DATASETS MAY BE LOST IF REPLY IS 'GO' ***
```

```
COPY OF ONE ALLOCATION UNIT COMPLETE num BLOCK COPIED
WITHOUT ERROR.  ERRORS ENCOUNTERED ON num BLOCKS.  ENTER
ANY REPLY TO REBUILD DAT AND CONTINUE.  DATASETS MAY BE
LOST.  DEVICE LABEL MAY BE RE-WRITTEN.
```

```
*** NOT ENOUGH SPACE ON DEVICE device-name TO REALLOCATE.
ENTER ANY REPLY TO TRY RE-INITIALIZING EXISTING BLOCK.
DATASETS MAY BE LOST.
```

```
FAILING BLOCK ON DEVICE device AT ALLOCATION UNIT num
   (CYLINDER CYL
    HEAD GROUP head
    SECTOR sector)
REPLY 'GO' TO ENTER AI INTO FLAW TABLE IN LABEL
REPLY 'SKIP' TO NOT ENTER AI INTO FLAW TABLE
```

If invalid or unexpected replies are entered, the messages are repeated with the following additional line:

```
text INVALID.  CORRECT AND RE-ENTER.
```

If Startup detects internal inconsistencies in the DSC (such as an entry being marked as a continuation but not contained in any dataset DSC chain) one or more of the following messages are issued. See the COS Message Manual, publication SR-0039, for further explanations.

```
DURING DSC RECOVERY num ENTRIES WERE ORPHANS.
REPLY 'GO' TO RELEASE ORPHANED DSC ENTRIES
REPLY 'IGNORE'  TO LEAVE ORPHANED ENTRIES IN USE
DURING DSC RECOVERY num ENTRIES WERE 'LOST'.  THIS IS A
SERIOUS CATALOG PROBLEM.  NOTIFY A SYSTEM ANALYST
IMMEDIATELY.  AN INSTALL SHOULD BE PERFORMED AS SOON
AS POSSIBLE.
REPLY 'GO' TO CONTINUE STARTUP
```

If invalid or unexpected replies are entered, the messages are repeated with the following additional line:

```
text INVALID.  CORRECT AND RE-ENTER.
```

5.5.3.6 Rolled job recovery processing

Following permanent dataset recovery, Startup optionally attempts to recover jobs that were in execution and have a valid roll image. This option is available only for a RESTART, and can be enabled or disabled through the *RRJ parameter file directive. If recovery of rolled jobs is enabled, Startup issues the following informative message:

```
ROLLED JOBS ARE BEING RECOVERED
```

5.5.3.7 System Directory processing

Following the recovery of rolled jobs, Startup optionally attempts to rebuild the SDR from a special recovery dataset, which is saved in the DSC as if it were a user dataset. This may be disabled via the *SDR parameter file directive. If enabled, Startup issues the following informative message:

```
THE SYSTEM DIRECTORY IS BEING RECOVERED
```

If disabled or during an INSTALL, Startup issues the following informative message:

```
CREATING NEW EDITION OF THE SYSTEM DIRECTORY
```

5.5.3.8 System dump processing

If a newly created system dump exists, Startup issues the following informative message:

```
THE SYSTEM DUMP IS BEING SAVED
```

The following error messages deal with problems encountered while trying to reserve the preallocated system dump area maintained by Startup.

The following error is issued when Startup is unable to read a track of the preallocated area.

```
CANNOT READ SYSTEM DUMP AREA
ENTER GO TO REALLOCATE AREA
ENTER RETRY TO RE-READ
```

The next message deals with the inability to reserve the preallocated area from the AI list contained in the dump header. The problem has occurred either because a flaw was added or because the system dump header had been corrupted.

```
SYSTEM DUMP AREA INVALID OR CONTAINS AI CONFLICT
ENTER GO TO REALLOCATE AREA
ENTER RETRY TO RE-READ EXISTING HEADER
```

Either one of the above problems could cause Startup not to create a new system dump from the information in the preallocated area.

The following errors occur when Startup is attempting to move the system dump information in the preallocated area to a permanent dataset. If Startup is unable to copy the information to the newly allocated dataset, the following message is displayed:

```
ERROR WRITING SYSTEM DUMP COPY
ENTER GO TO REALLOCATE AND RECOPY
ENTER RETRY TO TRY AND WRITE COPY AGAIN
ENTER QUIT TO ABANDON COPY OF DUMP
```

If, during the copy of the dump, the number of dump items exceeds the available space in the header, the following message is displayed:

```
TOO MANY DUMP ITEMS - SOME LOST
```

When attempting to save the newly created dump dataset, if PDM returns an error status, Startup issues the following message:

```
UNABLE TO SAVE COPY OF DUMP.  TYPE GO TO IGNORE.
```

After the system dump has been copied and saved as a permanent dataset, Startup has to rewrite the header in the preallocated area to show that the information in the preallocated area need not be copied. If an error occurs during the write, the following message is issued:

```
ERROR WHILE RE-WRITING DUMP HEADER
ENTER GO TO SKIP WRITE
ENTER RETRY TO TRY AGAIN
```

This error may cause duplicate copies of the same dump during subsequent Startups.

To enhance system dump processing, Startup (after determining that a system dump exists) queries the operator for a comment describing the dump as follows:

```
SYSTEM DUMP PROCESSING
ENTER COMMENT DESCRIBING REASON FOR TAKING SYSTEM DUMP
```

Startup requires a nonnull response and the comment is put in the system log and dump header.

5.5.3.9 DXT creation, validation, and recovery

Nonfatal error conditions involving the $DSC-EXTENSION dataset produce the following error messages that require a reply from the operator to continue processing. The required reply is contained within the messages, which follow in alphabetical order:

DD-XX-XX DEVICE IS EITHER UNAVAILABLE OR IS OFF. REPLY WITH THE (DXT,DV=,SZ=,OFV=,CAI=) COMMAND TO CONTINUE STARTUP.

DD-XX-XX DEVICE NAME NOT FOUND IN ANY EQT. REPLY WITH THE (DXT,DV=,SZ=,OFV=,CAI=) COMMAND TO CONTINUE.

DD-XX-XX HAS INSUFFICIENT CONTIGUOUS SPACE FOR THE $DSC-EXTENSION DATASET. REPLY WITH THE (DXT,DV=,SZ=,OFV=,CAI=) COMMAND TO CONTINUE STARTUP.

DD-XX-XX HAS INSUFFICIENT SPACE FOR THE $DSC-EXTENSION. REPLY WITH THE (DXT,DV=,SZ=,OFV=,CAI=) COMMAND TO CONTINUE STARTUP.

DD-XX-XX HAS NO SPACE. CANNOT START THE $DSC-EXTENSION ON IT. REPLY WITH THE (DXT,DV=,SZ=,OFV=,CAI=) COMMAND TO CONTINUE STARTUP.

DXT SIZE CANNOT BE ZERO. REPLY WITH THE (DXT,DV=,SZ=,OFV=,CAI=) COMMAND TO CONTINUE STARTUP.

INSUFFICIENT DISK SPACE FOR THE $DSC-EXTENSION. REPLY WITH THE (DXT,DV=,SZ=,OFV=,CAI=) COMMAND TO CONTINUE.

NO DEVICE HAS SUFFICIENT SPACE TO CONTAIN THE $DSC-EXTENSION. REPLY WITH THE (DXT,DV=,SZ=,OFV=,CAI=) COMMAND TO CONTINUE STARTUP.

PARAMETER FILE CONTAINS AN INVALID DXT COMMAND. TO CORRECT THE COMMAND, REENTER THE DXT COMMAND (DXT,SZ=,DV=,OVF=,CAI=).

PDN=*pdn*
ID=*id*
ED=*ed*
OWN=*owner*
DATASET HAS CATASTROPHIC DXT ERROR. ENTER (GO) TO CONTINUE OR (SKIP) TO CONTINUE W/O ANY FURTHER DXT ERROR REPORTING.

WARNING THE $DSC EXTENSION IS XX% FULL. THE SIZE OF THIS DATASET SHOULD BE INCREASED IN THE NEAR FUTURE VIA THE (DXT,SZ=+XXX) COMMAND. ENTER (GO) TO CONTINUE STARTUP.

TOO MANY BAD TRACKS WERE ENCOUNTERED WHILE TRYING TO INITIALIZE THE DXT ON DISK. THE BAD TRACKS HAVE BEEN FLAWED OUT. ENTER (GO) TO RETRY.

If Startup detects block number errors while reading the DXT, or if DQM detects errors while performing I/O on the DXT, Startup issues one or more of the following messages and waits for operator responses. Each message indicates appropriate responses. See the COS Message Manual, publication SR-0039, for more detailed explanations of causes and results of the various replies.

***** BLOCK NUMBER ERROR IN DXT *****
BLOCK EXPECTED = *blk* BLOCK FOUND = *blk*
REPLY 'GO' TO CLEAR - DATASETS MAY BE LOST
REPLY 'RETRY' TO RETRY OPERATION

*** DQM ERROR *errcode* WHILE *operation* DXT ***
REPLY 'GO' TO ATTEMPT REALLOCATION OF TRACK
REPLY 'RETRY' TO RETRY LAST REQUEST
*** DATASETS MAY BE LOST IF REPLY IS 'GO' ***

COPY OF ONE ALLOCATION UNIT COMPLETE *num* BLOCKS COPIED WITHOUT ERROR. ERRORS ENCOUNTERED ON *num* BLOCKS. ENTER ANY REPLY TO REBUILD DAT AND CONTINUE. DATASETS MAY BE LOST. DEVICE LABEL MAY BE RE-WRITTEN.

*** NOT ENOUGH SPACE ON DEVICE *device-name* TO REALLOCATE
ENTER ANY REPLY TO TRY RE-INTIALIZING EXISTING BLOCK.
DATASETS MAY BE LOST.

```
FAILING BLOCK ON DEVICE device AT ALLOCATION UNIT num
    (CYLINDER cyl
     HEAD GROUP head
     SECTOR sector)
REPLY 'GO' TO ENTER AI INTO FLAW TABLE IN LABEL
REPLY 'SKIP' TO NOT ENTER AI INTO FLAW TABLE
```

If you enter invalid or unexpected replies, the system repeats the messages with the following additional line:

text INVALID. CORRECT AND RE-ENTER.

5.5.3.10 Messages requiring no operator response

$DSC-EXTENSION processing produces the following messages that do not require a reply:

INITIATING THE $DSC-EXTENSION RECOVERY AND VALIDATION.

STARTUP SELECTED CREATION OF THE $DSC-EXTENSION.

THE $DSC-EXTENSION IS XX% FULL.

THE $DSC-EXTENSION WAS CREATED AND SAVED SUCCESSFULLY.

THE $DSC-EXTENSION WAS RECOVERED AND VALIDATED SUCCESSFULLY.

5.5.3.11 RDM informational messages

The following is a list of RDM informational messages that appear at STARTUP time:

WARNING -- RDM IS INACTIVE. *date time*

The RDM task has been inactivated by a system assembly value, the *RDMSTT startup directive, or an operator response to a problem detected. The *date* and *time* represent the RDM build time.

RDM INITIALIZATION COMPLETE. *date time*

The RDM task is active and ready to process a request. The *date* and *time* represent the build time.

5.5.3.12 RDM control variable message

The following RDM control variable messages appear at STARTUP time; an example is shown in figure 5-1. An explanation of its parameters follows.

```
                              ~stl/tmp/sm0043.rdmcnt

CRAY STATION.  VERSION 5.0.5. IOS.  L S R M              06/15/89  06:11:39

                    CRAY STATION MESSAGES -   INFORMATION ONLY
                                                                FRAME  0
MN    TIME                       MESSAGE
__    ____                       _______

 1    06:11   STARTUP IS PERFORMING A DEADSTART
 2    06:11   THE DATASET CATALOG IS BEING RECOVERED
 4    06:11  RDM INITIALIZATION COMPLETE.              06/14/89  11:52:00
 5    06:11   RDM TASK STATUS INFORMATION.
               LOG =        6    ERROR   TRACE

 6    06:11    COS    DISK     JOB     JOB  PASSWORD EXPIRE  SITE P1 SITE P2
             SECURITY CONTROL  CHARGE CONTROL  CHECK PASSWORD
             SEC=FULL DCO= ON CHG= ON JOB= ON PCO= ON PEO= ON     OFF     OFF
 7    06:11                                                  SITE P3 SITE P4

                                                                  OFF     OFF
```

Figure 5-1. Example of RDM Variable Message

COS	DISK	JOB	JOB	PASSWORD	EXPIRE	SITE P1	SITE P2
SECURITY	CONTROL	CHARGE	CONTROL	CHECK	PASSWORD		
SEC=*sec*	DCO=*dco*	CHG=*chg*	JOB=*job*	PCO=*pco*	PEO=*peo*	SITE P3	SITE P4

SEC=*sec* — COS security as set by the system assembly or changed by the *RDMSTT startup directive. Values are as follows:

- OFF — COS Security is turned off.
- WARN — COS Security is in warn mode.
- ON — COS Security is active and will be enforced by RDM.

DCO=*dco* — Disk control of user limits for permanent and archive dataset as set in the RD Information entry or changed by the *RDMSTT directive. Values are as follows:

- OFF — Disk control is turned off.
- ON — Disk control is active and will be enforced by RDM.

CHG=*chg* — Job charging for user jobs as set by system assembly or changed by the *RDMSTT directive. Values are as follows:

OFF	Job charging is turned off.
ON	Job charging is active and will be enforced by RDM.

JOB=*job* — Job control of user limits for JOB statement parameters as set in the RD information entry or changed by the *RDMSTT startup directive. Values are as follows:

OFF	Job control is turned off.
ON	Job control is active and will be enforced by RDM.

PCO=*pco* — Password checking of passwords on the ACCOUNT statement as set in the RD Information entry or changed by the *RDMSTT startup directive. Values are as follows:

OFF	Password checking is turned off.
ON	Password checking is active and will be enforced by RDM.

PEO=*peo* — Password expiration for user passwords as set in the RD Information entry or changed by the *RDMSTT startup directive. Values are as follows:

OFF	Password expiration is turned off.
ON	Password expiration is active and will be enforced by RDM.

5.5.4 STARTUP SUBMITTED JOB PROCESSING

Nonfatal error conditions resulting from submitting jobs via the *SUBMIT parameter file directive during system startup produce the following error messages that require a reply from the operator to continue processing. The required reply is contained within the messages, which follow:

UNABLE TO OBTAIN SDT ENTRY

UNABLE TO OBTAIN DAT ENTRY

NO READ PERMISSION

DATASET NOT FOUND

PDM ERROR CODE *nnn* ON ACCESS REQUEST

DATASET HAS BEEN PARTIALLY DELETED

DQM ERROR CODE *nnn* ON READ REQUEST

DQM ERROR CODE *nnn* ON WRITE REQUEST

```
   PDM ERROR CODE nnn ON SAVE REQUEST

   STARTUP SUBMIT ERROR ON DATASET dataset
     OWN = owner  ID = id  ED = nn
```

(one of the above error messages goes here)

```
   ENTER:  ABORT     TO TERMINATE STARTUP AT THIS POINT
                   END
   TO IGNORE REMAINING SUBMIT DIRECTIVES
                   SKIP
TO TRY THE NEXT SUBMIT DIRECTIVE
                   RETRY
TO RETRY SUBMITTING THIS DATASET
```

5.5.5 FAST SECONDARY STORAGE (FSS) RESTORATION

If volatile devices are configured in the EQT and a *RESTORE parameter file directive was encountered for one or more of them, the following informative message is issued for each such device:

```
   DEVICE name IS BEING RESTORED
```

I/O errors during the process of restoring FSS devices (the SSD and Extended Buffer Memory) cause the sector in error to be marked as bad and any dataset residing on that area is not recovered. The following message is displayed when such an error occurs:

```
   UNABLE TO RESTORE INFORMATION ON DEVICE device
   ERROR OCCURRED DURING WRITE ON
               CYLxx,TRKxx,SECTxx
   TO CONTINUE TYPE - GO
```

If the system has been recovered with any FSS configured up, the associated backed-up information is marked as invalid. Should the site then attempt to restore that FSS device without performing another flush, Startup informs the operator that the information might be invalid and gives the option of using the backed-up information. The warning message is as follows:

```
   THE BACKED UP INFORMATION FOR DEVICE ldv
   MAY NOT BE VALID

   IF THE RESTORE PROCESS IS TO CONTINUE
   WITH THIS INFORMATION
               TYPE - GO
```

```
IF THE DEVICE IS TO BE RECOVERED
WITHOUT ANY RESTORE PROCESSING
            TYPE - SKIP
```

If during a deadstart, warmstart, or restart, Startup is unable to find a label on an FSS-type device and no corresponding *RESTORE directive is included in the parameter file, the following message is displayed:

```
NO LABEL WAS FOUND ON DEVICE device
INFORMATION CONTAINED ON THIS DEVICE MAY BE INVALID

TO PROCEED WITH STARTUP AND REWRITE THE DEVICE LABEL
    TYPE -- CONTINUE

TO PROCEED WITH STARTUP WITH THIS DEVICE DOWN
    TYPE -- DOWN

TO PROCEED WITH STARTUP AND REWRITE THE DEVICE LABEL
AND RESTORE THE INFORMATION ON THE DEVICE
    TYPE -- RESTORE
```

Even if label rewrite is forced (*CONFIG, ... ,WDL=Y), the message tells the operator the information on that device is bad. If a *RESTORE directive is included for a device and startup is unable to restore the information on that device, the following message is displayed:

```
CAN NOT RESTORE INFORMATION ON VOLATILE DEVICE device
BECAUSE reason
TYPE - GO TO CONTINUE STARTUP WITHOUT RESTORING THIS DEVICE
TYPE - DOWN TO CONTINUE STARTUP WITH THE DEVICE CONFIGURED DOWN
```

The reasons for the previous messages being displayed are as follows:

```
NO BACKUP PERMANENT DATASET EXISTS

NO DATA EXISTS ON DATASET

UNABLE TO READ BACKED UP DATASET

OF SIZE DIFFERENCE BETWEEN BACKUP DATASET AND DEVICE
```

If a device is declared volatile, EQVOL is set. If Startup determines that problems exist with the back-up dataset associated with this device, the following message is displayed:

```
THE BACKUP DATASET DN=$device reason
TYPE - GO
TO CONTINUE STARTUP WITH THIS DEVICE NOT MARKED AS VOLATILE
```

TYPE - NEW
TO CREATE A NEW EDITION OF THIS DATASET

The reasons for the previous messages being displayed are as follows:

DATASET IS NOT LARGE ENOUGH TO HOLD ALL THE INFORMATION CONTAINED ON THIS VOLATILE DEVICE

CONTAINS I/O ERROR
STARTUP IS UNABLE TO READ/WRITE THE HEADER INFORMATION

6. STATION DEBUG COMMANDS

The Data General Station (DGS) and the I/O Subsystem (IOS) station provide the system analyst with a set of debug commands and displays in addition to the features described in the Data General Station (DGS) Operator's Guide, publication SG-0006, and in the I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051. See the I/O Subsystem (IOS) Administrator's Guide, publication SG-0307, for complete information on these commands.

7. DUMPING THE CRAY COMPUTER SYSTEM

This section outlines the different dump options and formats for interpreting the dump following a system failure. The first subsection outlines dump methods for different configurations, and the second subsection describes different formats used to process the system dumps.

7.1 DUMPING THE CRAY COMPUTER SYSTEM

Following a system failure, the operator has the option of dumping the Cray computer system. This can be done in any of the following ways, depending on the system configuration:

- Dump to the Data General Station (DGS) Eclipse disk
- Dump to the Cray disk by the DDC utility
- Dump to the Cray disk using the I/O Subsystem (IOS) SYSDUMP procedure
- Dump to printer using the DMP utility at the IOS

7.1.1 DUMPING THROUGH A DATA GENERAL STATION

The station software running on the DGS Eclipse connected to the Cray Maintenance Control Unit (MCU) channel provides the operator with the ability to dump selected portions of Cray Central Memory to the disk pack associated with that Eclipse. When the dump completes transferring memory to the Eclipse disk, the operator can issue a START command to reinitialize COS in the Cray Central Memory. The operator has the option of immediately printing the dump using DMP, a utility executing on the Eclipse, or staging it into COS, where it can be made a permanent dataset and can be printed by the system utility FDUMP. The dump is invoked with the DUMP command. The resulting dump dataset can be saved if the SAVE command follows the DUMP command.

Format:

```
DUMP,filename,fwa,lwa
```

filename DGS file name associated with the dump

fwa Beginning Cray Central Memory address (absolute)

lwa Ending Cray Central Memory address (absolute)

To send the dump to be saved as a Cray permanent dataset, use the following command:

Format:

```
SAVE,filename,pdn
```

filename DGS file name associated with the dump

pdn Name of Cray permanent dataset to be created by COS

Once the dump resides on the Cray disk, it can be accessed by a user job printing selected portions of the dump using the system utility FDUMP.

7.1.2 DUMPING TO CRAY DISK STORAGE

The DDC utility dumps memory directly from the Central Memory to the Cray disk storage units. The master device must be attached directly to the Cray mainframe; it cannot be attached to the IOS. The device label for the master device must contain a pointer to the first track of a disk area preallocated to hold a system dump. The first sector of the track must contain the list of all tracks reserved for the dump. A block of words is reserved in EXEC so the program can be loaded and executed without overwriting EXEC code or tables.

The DDC utility is invoked by using the SYSDUMP command at a Data General station connected to the MCU channel. The absolute binary of the DDC program must reside on the MCU in unblocked format.

Format:

```
 ___________
|           |
|  SYSDUMP  |
|___________|
```

Memory is dumped from 0 to C@MMSIZE, or until the reserved disk space is filled. A flag is set in the first sector (word 511) indicating a new dump exists. During the next Startup, the flag is recognized and a copy of the preallocated area is made and saved as a permanent dataset. The flag is then cleared. This permanent dataset can be accessed by user jobs to print selected portions.

Normally, the permanent dataset created by Startup is called CRAY1SYSTEMDUMP; this can be changed by the installation. In response to certain errors returned by the Permanent Dataset Manager (PDM), Startup changes the name of the dataset and tries to save it again. Errors handled in this way include TOO MANY EDITIONS and MAINTENANCE PERMISSION NOT GRANTED. The log message issued when the dataset is successfully saved identifies the name by which the dataset is known to the dataset catalog (DSC).

When it is necessary to change the name, Startup locates the last nonzero character in the permanent dataset name (PDN) and increments it by 1. If the resulting character is alphanumeric, the old name is replaced and the SAVE is attempted again. If the resulting character is not alphanumeric, the original character is left alone and Startup moves one character to the left and repeats the process. If no character in the name can be incremented by 1 without producing a nonalphanumeric character, Startup is unable to save the dataset and a system log message is issued to that effect.

7.1.3 DUMPING TO DISK USING THE IOS

The IOS SYSDUMP procedure dumps information from selected components of the Cray computer system to the master device. The IOS must be connected to the MCU channel; the master device must be attached to the IOS Buffer I/O Processor (BIOP). The master device must contain the same allocation information required by DDC.

Initiate the SYSDUMP procedure by pressing CONTROL-D at the Master I/O Processor (MIOP) Kernel console. During the dialogue with SYSDUMP, the operator can select areas of the Cray computer system to dump. For a description of this interaction, see the I/O Subsystem (IOS) Operator's Guide for COS. SYSDUMP execution requires some basic functions (the Kernel software) to be operational in the IOS; if SYSDUMP does not execute, the operator can resort to the IOS DMP utility.

The dump dataset created by SYSDUMP is copied to a permanent dataset by Startup as described earlier in this section.

7.1.4 DUMPING TO THE IOS PRINTER

DMP is a stand-alone utility program that executes in the IOS and dumps selected areas of the Cray computer system to the printer. DMP is loaded into the IOS from tape or an 80-Mbyte disk and interacts with the operator to determine the dump content. The I/O Subsystem (IOS) Operator's Guide for COS, publication SG-0051, fully describes the DMP deadstart operation, dialogue, and output.

As for SYSDUMP processing, the IOS must be attached to the Cray mainframe through the MCU channel.

7.2 SYSTEM DUMP FORMAT

System dump datasets created by the DUMP, DDC utility, and IOS SYSDUMP procedures are in unblocked format, and cannot be read by the Fortran library blocked I/O routines. The utility FDUMP processes these dumps. (See the Operational Aids Reference Manual, publication SM-0044, for a complete description of the FDUMP utility.)

7.2.1 FORMAT FROM DATA GENERAL DUMP

When a dump is staged in from the Data General ECLIPSE using the following sequence, the permanent dataset created is an unblocked binary dataset, where word 0 of the dataset represents *fwa* and the last word of the dataset represents *lwa*, as specified on the DUMP command. The dataset contains no headers or trailers. This dataset can be printed using the FDUMP utility.

Formats:

DUMP,*filename*,*fwa*,*lwa*
SAVE,*filename*,*pdn*

7.2.2 FORMAT FROM DUMP TO DISK STORAGE UNITS

When the COS utility DDC or the IOS SYSDUMP procedure is used to create the deadstart dump, the permanent dataset created is an unblocked dataset with the following format:

Word Address	Contents
0-511	Dump header
512-end	Mainframe memory dump
	CPU registers
	Buffer Memory dump
	IOP information

The dump header is a 512-word block containing information describing the data captured in the remainder of the dataset. The DDC utility captures only the Central Memory dump and CPU registers; the IOS SYSDUMP supplies the other information. See the FDUMP description in the Operational Aids Reference Manual, publication SM-0044, for a detailed description of the dump dataset.

APPENDIX SECTION

A. INSTALLATION PARAMETER LISTING

This appendix is divided into COS installation parameters and I/O Subsystem (IOS) installation parameters.

A.1 COS INSTALLATION PARAMETERS

COS installation parameters are in three groups: hardware configuration parameters, target machine parameters, and operating system parameters.

A.1.1 HARDWARE CONFIGURATION PARAMETERS

Hardware configuration parameters are all equates in deck CONFIG@P, which is called by both COSTXT/COSDEF and $SYSTXT/$SYSDEF. Five hardware configuration parameter categories are defined: CPU, additional CPU, Central Memory, SSD solid-state storage device, and Disk Storage Unit (DSU). These parameters always start with the characters C@ and take an integral value or a type. Types are defined in the deck CONFIG@P and start with the character @.

If a CPU other than the host CPU has been targeted in CONFIG@P, the installation parameters must be defined in CONFIG@M. These parameters always begin with the characters M@ and are listed in the Target CPU Parameter section of this manual.

The following is a list of the hardware parameters. All numeric default values are given in decimal unless otherwise indicated. If more than one default value is given, then conditional assembly is used to select the default.

A.1.1.1 CPU parameters

Parameter	Default	Significance
C@CPAVL	0	Additional vector logical (AVL) availability. This option is disabled by default because it is not present on some older CRAY X-MP computer systems, and there are some codes that may run slower if it is enabled. Consult the hardware reference manual for specific details. The parameter values and their descriptions follow: 0 AVL unit is not available 1 AVL unit available (some CRAY X-MP computer systems only)
C@CPCIGS	0	Compress index, gather/scatter (CIGS) availability. CIGS was optional on some older model CRAY X-MP mainframes. The parameter values and their descriptions follow: 0 CIGS is not available 1 CIGS is available
C@CPCYCL	9497	CPU cycle time in picoseconds. The parameter values and their descriptions follow: 5998 CRAY Y-MP with 6 nanosecond clock 8496 CRAY X-MP system with 8.5 nanosecond clock 9497 CRAY X-MP computer system with 9.5 nanosecond clock 10000 CRAY X-MP/14se system 12000 CRAY-1 M system 12500 CRAY-1 A, B, or S system
C@CPEMA	0	Extended memory addressing (EMA) availability. EMA must be enabled if your mainframe has the associated hardware (newer CRAY X-MP mainframes have EMA hardware regardless of the amount of central memory with which they are shipped). For the CRAY Y-MP, C@CPEMA must be set. The actual use of EMA is controlled by the M@EMA parameter described in subsection A.1.2, Target CPU Parameters. The parameter values and their descriptions follow: 0 EMA is not available 1 EMA is available (some CRAY X-MP mainframes only)

Parameter	Default	Significance
C@CPHCHN	C@CPRCHN+1	Highest real or pseudo-channel number (always an odd number). This parameter is equal to the highest channel number (C@CPRCHN) if there are no pseudo-channels.
C@CPHPG	3	Highest hardware performance monitor group (if present)
C@CPLCHN	2, 8, or 16	Lowest channel number (always an even number). The parameter values and descriptions follow: 2 CRAY-1 A, B, S, or M system 8 CRAY X-MP system 16 CRAY Y-MP system
C@CPMCHN	2, 8, or 16	MCU input channel number (always an even number). The parameter values and descriptions follow: 2 CRAY-1 A, B, S, or M system 8 CRAY X-MP system 16 CRAY Y-MP system
C@CPPCHN	C@CPRCHN+1	First pseudo-channel. This parameter is used for the GOS station and is always one greater than the maximum real channel number (always even).
C@CPQUAN	1	Number of processors (CPUs). The parameter values and their descriptions follow: 1 CRAY-1 A, B, S, M, or CRAY X-MP/1 system 2 CRAY X-MP/2 system 4 CRAY X-MP/4 system 8 CRAY Y-MP system
C@CPRCHN	15, 23, 25, or 31	Highest physical channel number (always odd). This parameter is equal to the highest channel number (C@CPRCHN) if there are no pseudo-channels. The following are the parameter values and their descriptions: 15 CRAY X-MP system 23 CRAY X-MP EA system 25 CRAY-1, A, B, S, or M system 31 CRAY Y-MP system

Parameter	Default	Significance
C@CPSTR	0 or 1	Status register (STR) availability. The parameter values and their descriptions follow: 0 STR is not available (CRAY-1 A, B, S, or M system 1 STR is available (CRAY X-MP system)
C@CPSUBT	@XMP3XX	CRAY X-MP host CPU subtype (valid only if C@CPTYPE is equal to @CRAYXMP. The parameter values and their descriptions follow: @XMP101 (8) CRAY X-MP/2 CPU (s/n 101-114) @XMP1XX (9) CRAY X-MP/2 CPU (s/n 115+) @XMP2XX (10) CRAY X-MP/4 CPU (s/n 201+) @XMP3XX (11) CRAY X-MP/1 CPU (s/n 301+) @XMP5XX (12) CRAY X-MP 14se (s/n 501+) @YMP11XX (13) CRAY X-MP EA/464(s/n 1100+) @YMP10XX (14) CRAY Y-MP (s/n 1000+)
C@CPTSUB	C@CPSUBT	CRAY X-MP Host Target subtype (valid only if C@CPTYPE is equal to @CRAYXMP. From the set: @XMP101 (8) CRAY X-MP/2 CPU (s/n 101-114) @XMP1XX (9) CRAY X-MP/2 CPU (s/n 115+) @XMP2XX (10) CRAY X-MP/4 CPU (s/n 201+) @XMP3XX (11) CRAY X-MP/1 CPU (s/n 301+) @XMP5XX (12) CRAY X-MP 14se (s/n 501+) @YMP11XX (13) CRAY X-MP EA/464(s/n 1100+) @YMP10XX (14) CRAY Y-MP (s/n 1000+)
C@CPTARG	C@CPTYPE	CPU Target Type. The parameter values and their descriptions follow: @CRAY1 (1) CRAY-1 A or B system @CRAY1S (2) CRAY-1 S system @CRAYXMP (3) CRAY X-MP system @CRAY1M (4) CRAY-1 M system
C@CPTYPE	@CRAYXMP	CPU host type. The parameter values and their descriptions follow: @CRAY1 (1) CRAY-1 A or B system @CRAY1S (2) CRAY-1 S system @CRAYXMP (3) CRAY X-MP system @CRAY1M (4) CRAY-1 M system

Parameter	Default	Significance
C@MODEL	@XMP22	Machine model type. Used to determine correct location of the chip select in the Exchange Package. The parameter values and their descriptions follow:

@XMP11	(10)	CRAY X-MP/11
@XMP12	(11)	CRAY X-MP/12
@XMP14	(12)	CRAY X-MP/14
@XMP14SE	(13)	CRAY X-MPse/14
@XMP18	(14)	CRAY X-MP/18
@XMP116	(15)	CRAY X-MP/116
@XMP21	(20)	CRAY X-MP/21
@XMP22	(21)	CRAY X-MP/22
@XMP24	(22)	CRAY X-MP/24
@XMP28	(23)	CRAY X-MP/28
@XMP216	(24)	CRAY X-MP/216
@XMP42	(40)	CRAY X-MP/42
@XMP44	(41)	CRAY X-MP/44
@XMP48	(42)	CRAY X-MP/48
@XMP416	(43)	CRAY X-MP/416
@XMP464E	(D'1100)	CRAY X-MP EA/464
@XMP432E	(D'1101)	CRAY X-MP EA/432
@XMP416E	(D'1102)	CRAY X-MP EA/416
@XMP164E	(D'1200)	CRAY X-MP EA/164
@XMP132E	(D'1201)	CRAY X-MP EA/132
@XMP116E	(D'1202)	CRAY X-MP EA/116
@XMP116S	(D'1300)	CRAY X-MPse EA/116
@XMP14S	(D'1301)	CRAY X-MPse EA/14
@YMP832	(D'1000)	CRAY Y-MP/832

A.1.1.2 Additional CPU parameters

Parameter	Default	Significance
C@AREG	24	Size of A registers in bits
C@BREG	24	Size of B registers in bits
C@CLSIZE	17	CRAY X-MP cluster size in words
C@CSZ	16 or 32	Size of instruction buffer in words. The parameter values and their descriptions follow: 16 CRAY-1 A, B, S, or M system 32 CRAY X-MP system

Parameter	Default	Significance
C@EXECMD	0 or 1	Indicates EXEC is to run in 32-bit addressing mode, if set. If zero, EXEC runs in XMP compatible mode. On a CRAY Y-MP system, this parameter must be set. This parameter is relevant only if C@YMP is nonzero.
C@HPM	0 or 1	Hardware performance monitor (HPM) availability. The parameter values and their descriptions follow: 0 HPM is not available (CRAY-1 A, B, S, or M system) 1 HPM is available (CRAY X-MP system)
C@L2BAB	4, 5, 6, or 8	Shift count for Base Address (BA) register. The parameter values and their descriptions follow: 4 CRAY-1 A, B, S, or M system 5 CRAY X-MP/1, CRAY X-MP/14se, or CRAY X-MP/2 system 6 CRAY X-MP/4 or CRAY X-MP EA/464 system 8 CRAY Y-MP system
C@L2CSZ	4 or 5	Shift count for instruction buffer. The parameter values and their descriptions follow: 4 CRAY-1 A, B, S, or M system 5 CRAY X-MP/1, CRAY X-MP/14se, or CRAY X-MP/2 system
C@L2SREG	6	Log (base 2) of size of S registers (C@SREG)
C@NAREG	8	Number of A registers
C@NBREG	64	Number of B registers
C@NSREG	8	Number of S registers
C@NTREG	64	Number of T registers
C@NUMCL	0, 3, 5, or 9	Number of clusters. The parameter values and their descriptions follow: 0 No clusters; CRAY-1 A, B, S, or M system 3 CRAY X-MP/1, CRAY X-MP/14se, or CRAY X-MP/2 system 5 CRAY X-MP/4 system 9 CRAY Y-MP system
C@NUMSB	8	Number of CRAY X-MP computer system shared B registers

Parameter	Default	Significance
C@NUMSM	32	Number of CRAY X-MP computer system semaphore bits
C@NUMST	8	Number of CRAY X-MP computer system shared T registers
C@PC	0 or 1	Programmable clock (PC) availability. The parameter values and their descriptions follow: 0 PC is not available (CRAY-1 A or B mainframe 1 PC is available (CRAY-1 S or M, or CRAY X-MP mainframe
C@SN	300	Mainframe serial number
C@SREG	64	Size of the S registers in bits
C@TREG	64	Size of the T registers in bits
C@VPOP	0 or 1	Vector population count (VPOP) availability. The parameter values and their descriptions follow: 0 VPOP is not available (some CRAY-1 A or B systems) 1 VPOP is available (CRAY-1 S or M, or CRAY X-MP system)
C@YMP	0 or 1	Indicates machine has 32-bit, A-register addressing, if set. The parameter values and their descriptions are as follows: 0 All other machine types 1 CRAY Y-MP or CRAY X-MP EA/464 system

A.1.1.3 Central Memory parameters

Parameter	Default	Significance
C@BBSY	4 or 8	Memory bank busy time in clock periods. The parameter values and their descriptions follow: 4 CRAY-1 A, B, S, or CRAY X-MP system with bipolar memory 8 CRAY-1 M or CRAY X-MP system with MOS memory
C@BLKSZ	512	Default block size in words. This parameter is used by CIO when reading or writing mass storage.

Parameter	Default	Significance
C@CHIPSZ	14	Log (base 2) of memory chip size (C@MMCHIP)
C@L2BLSZ	9	Log (base 2) of default block size for mass memory (C@BLKSZ)
C@MLOT	MLOTM10	Central Memory layout (used by EXTRACT for memory error reporting). The parameter values and their descriptions follow: MLOT1AB (1) CRAY-1 A or B system MLOT1S (2) CRAY-1 S system MLOTX101 (8) CRAY X-MP/2 (s/n 101-114) system MOLT1M19 (4) CRAY-1 M system MLOTX115 (9) CRAY X-MP/2 (s/n 115+) system MLOTM10 (11) CRAY X-MP/1 (s/n 301+) or CRAY X-MP/14se system MLOTX3XX (11) CRAY X-MP/1 (s/n 301+) or CRAY X-MP/14se system MLOTX201 (10) CRAY X-MP/4 (s/n 201+) system MLOTX2XX (10) CRAY X-MP/4 (s/n 201+) system MLOTX4XX (32) CRAY X-MP/2 system (64K memory chips, 8 columns) MLOT5XX (12) CRAY X-MP/14se (s/n 500+) MLOTXEA (13) CRAY X-MP EA/464 MLOTY32 (14) CRAY Y-MP/832
C@MMBANK	16	Number of banks in Central Memory
C@MMCHIP	@M16KCH	Central Memory chip size. The parameter values and their descriptions follow: @M1KCH (1) 1024-bit chips @M2KCH (2) 2048-bit chips @M4KCH (3) 4096-bit chips @M16KCH (4) 16,384-bit chips @M64KCH (5) 65,536-bit chips @M256KCH (6) 262,144-bit chips

Parameter	Default	Significance
C@MMCONF	@MBOTH	Central Memory configuration. The parameter values and their descriptions follow: @MLEFT (1) Left half (CRAY-1 systems only) @MRIGHT (2) Right half (CRAY-1 systems only) @MBOTH (3) Both halves @MUPPER (4) Upper half (CRAY X-MP system @MLOWER (5) Lower half (CRAY X-MP system)
C@MMSIZE	4000000_8	Central Memory size in words (4000000_8 = 1 Mword). This parameter must be less than or equal to the actual memory size. To obtain the Central Memory size for CRAY X-MP mainframes, subtract one word for each memory bank from the total size. If the mainframe has a full complement of memory, then the last word in each bank is not addressable.
C@MSPD	11, 13, or 14	Memory speed in clock periods. The parameter values and their descriptions follow: 11 CRAY-1 A, B, or S system 13 CRAY-1 M system 14 CRAY X-MP system with bipolar memory 17 CRAY X-MP system with MOS memory 17 CRAY Y-MP system

A.1.1.4 SSD solid-state storage device parameters

Parameter	Default	Significance
C@SSDCT	@NOSSD	SSD channel type. The parameters and their descriptions follow: @NOSSD (0) No SSD configured @SSDHSP (1) High-speed (HSP) channel (100-Mbyte) @SSDVHSP (2) Very-high-speed (VHSP) channel (1250-Mbyte)

Parameter	Default	Significance
C@SHSCCC	25	SSD HSP command channel number (always an odd number). This parameter is ignored unless C@SSDCT=@SSDHSP.
C@SHSCSC	24	SSD HSP status channel number (always an even number). This parameter is ignored unless C@SSDCT=@SSDHSP.
C@NSVHSP	2	Number of SSD VHSP channels. This parameter is ignored unless C@SSDCT=@SSDVHSP.
C@SVHSP0	1 or 7	SSD VHSP channel 0's channel number. This parameter is ignored unless C@SSDCT=@SSDVHSP. The parameter values and their descriptions follow: 1 CRAY Y-MP system 7 CRAY X-MP system
C@SVHSP1	5 or 6	SSD VHSP channel one. This parameter is ignored unless C@SSDCT=@SSDVHSP. The parameter values and their descriptions follow: 5 CRAY Y-MP system 6 CRAY X-MP system
C@SSDMTR	@TRHSP	SSD maximum transfer rate in Mbytes per second. This parameter is ignored if C@SSDCT=@NOSSD. The parameter values and their descriptions follow: @TRHSP (100) HSP channel @TRSD8 (320) VHSP channel, 8-Mword SSD @TRSD16 (640) VHSP channel, 16-Mword SSD @TRSD32 (1250) VHSP channel, 32-Mword SSD @TRSD128 (1250) VHSP channel, 128-Mword SSD @TRSD256 (1250) VHSP channel, 256-Mword SSD @TRSD512 (1250) VHSP channel, 512-Mword SSD
C@SSDBCA	0	SSD backdoor channel availability. This parameter indicates whether a HSP channel connects the IOS directly to the SSD. The parameter values and their descriptions follow: 0 SSD backdoor channel is not available 1 SSD backdoor channel is available

A.1.1.5 Disk Storage Unit (DSU) parameters

Parameter	Default	Significance
C@DDTRAN	925	DD-19/DD-29 typical 1-sector transfer time in microseconds
C@DDLATC	16600	DD-19/DD-29 maximum latency time in microseconds
C@DDTOUT	625000	DD-19/DD-29 seek time-out in microseconds
C@NSEBM	6912	Number of sectors of Buffer Memory used for Buffer Memory Resident (BMR) datasets (IOP sites only). The lower portion of Buffer Memory is used by IOS and the upper portion allocated for BMR datasets. This parameter should be a multiple of 128, since Buffer Memory is allocated between COS and IOS in 65K word blocks.

A.1.2 TARGET CPU PARAMETERS

Target CPU parameters are all equates in deck CONFIG@M, which is called by both COSTXT/COSDEF and $SYSTXT/$SYSDEF. These parameters always start with the characters M@ and take a numeric value or are equated to another symbol - usually from CONFIG@P.

Target CPU parameters defined in CONFIG@M are released with values that are essentially equal to the host parameters defined in CONFIG@P. If the target and host CPU types are not equal, these parameters must be set in CONFIG@M. The parameters are as follows:

Parameter	Default	Significance
M@AVL	C@CPAVL	Additional vector logical unit availability
M@BBSY	C@BBSY	Memory bank busy time in clock periods
M@CIGS	C@CPCIGS	Compress index, gather/scatter availability
M@CPCYCL	C@CPCYCL	CPU cycle time in picoseconds
M@CPSUBT	C@CPSUBT	CRAY X-MP computer system host CPU subtype
M@CPTYPE	C@CPTYPE	CPU host type

Parameter	Default	Significance
M@EMA	0 or C@CPEMA	Extended memory addressing (EMA) hardware availability. Defaults to zero if C@MMSIZE is less than 4 Mwords; otherwise, it defaults to C@CPEMA.
M@HOST	1	Target machine is the host by default. The parameter values and their descriptions follow: 0 Target machine is not the host machine 1 Target machine is the host machine
M@HPM	C@HPM	Hardware Performance Monitor availability
M@MMBANK	C@MMBANK	Number of banks in Central Memory
M@MMSIZE	C@MMSIZE	Central Memory size in words
M@MSPD	C@MSPD	Memory bank speed in clock periods
M@NUMCL	C@NUMCL	Number of clusters
M@PC	C@PC	Programmable clock availability
M@SN	C@SN	Cray mainframe serial number
M@STR	C@STR	Status register availability
M@VPOP	C@VPOP	Vector population count availability

A.1.3 OPERATING SYSTEM PARAMETERS

Operating system parameters are equates or constants. Equates are in the deck COSI@P and USERI@P called by COSTXT/COSDEF. Constants are in the deck STPTAB, having a Cray Assembly Language (CAL) IDENT and list option of STPTAB.

Type indicates how the parameter is defined. Those defined by equates in COSTXT/COSDEF are flagged with CD, those defined by equates in COSTXT/COSDEF using the RGNAME macro are flagged with CD*, those defined by equates in STP are flagged with SD, and those defined by CON pseudo instructions in STP are flagged with SV. (These CON pseudo-ops are in UPDATE deck STPTAB, and the CAL IDENT of the data subroutine is also STPTAB.) Default indicates the released (default) value chosen for the average new site. All values are decimal unless otherwise indicated.

When setting time values, use the $CYCLES macro as described in the Macros and Opdefs Reference Manual, publication SR-0012.

The parameters are arranged in groups defined by their function within the operating system. These parameter groups are the following:

- Account
- Job Scheduler (JSH)
- Dataset Management (DM)
- Startup
- System
- Security
- Disk
- Tape
- Station
- System Performance Monitor (SPM)
- Interactive
- FSS preemption (FSS)
- User
- SUPERLINK MVS (SL/MVS)
- Resource Dataset Management (RDM)

Table A-1 is an alphabetic listing of the operating system parameters.

Table A-1. Operating System Parameters - Alphabetic

Operating System Parameter	Function
I@$INSZ	User
I@$ODLM	User
I@$OMLM	User
I@ACT	Account
I@ALLSDT	Station
I@APLN	Station
I@AUTOFL	User
I@AVL	User
I@AVR	Tape
I@BDM	User
I@BFDECR	System
I@BFI	User
I@BFIDLE	System
I@BFINCR	System
I@BFSIZE	System
I@BPL	Tape
I@BRBGN	System
I@BRF	Account
I@BRNUM	System
I@BSF	Account
I@BULLIT	User
I@CECT	System
I@CEDDI	System
I@CHATIM	System
I@CLC	Account
I@COSMIN	System
I@CPAGBI	Disk
I@CPDSK	Disk
I@CPFSS	Disk
I@CRYPT	Security
I@CSDMAX	JSH
I@DD	Disk
I@DDCCH	System
I@DDCUN	System
I@DEFINT	System
I@DEFLM	DM
I@DLPP	RDM
I@DMI	Account
I@DMPCMP	System
I@DMPSIZ	System
I@DNBFZ	User
I@DNDT1	User
I@DNDT2	User

Operating System Parameter	Function
I@DNDT3	User
I@DNOVF	User
I@DNPDMO	User
I@DNSPD	User
I@DNSZ	User
I@DNXMAX	User
I@DNXMIN	User
I@DPAM	RDM
I@DPWAIT	User
I@DTDBS	Tape
I@DTDREP	DM
I@DTRDLY	Station
I@DVLRES	System
I@DXTCAI	Disk
I@DXTFUL	DM
I@DXTLDV	DM
I@DXTOVF	DM
I@DXTSZ	DM
I@EFI	User
I@EMA	User
I@EXPANS	JSH
I@FDPMAX	DM
I@FEMSK	User
I@FIT	System
I@GOSPWS	System
I@GOSSIZ	System
I@GOUMIN	System
I@GROUP	Station
I@IACTR	Interactive
I@IAIBT	Interactive
I@IAPOLL	Station
I@IJTL	User
I@IMXTXT	Station
I@INSMX	Station
I@INST	Station
I@INTERN	Station
I@IOB	Account
I@IOF	Account
I@IOPICH	System
I@IOR	Account
I@JCCHAR	JSH
I@JFLMAX	User
I@JFLMSG	User

Table A-1. Operating System Parameters - Alphabetic (continued)

Operating System Parameter	Function
I@JNEMLM	JSH
I@JOBMIN	JSH
I@JSCOS	JSH
I@JSHTLE	JSH
I@JSITS	JSH
I@JSLK1	JSH
I@JSLK2	JSH
I@JSLK3	JSH
I@JSLK4	JSH
I@JSMPA	JSH
I@JSMPB	JSH
I@JSMPC	JSH
I@JSMPD	JSH
I@JSRRI	JSH
I@JSTEI	JSH
I@JSTS0	JSH
I@JSTS1	JSH
I@JSTS2	JSH
I@JSTS3	JSH
I@JTLDEF	JSH
I@JXTSIZ	JSH
I@LBMWL	Tape
I@LGBSZ	System
I@LGDSZ	System
I@LGUSZ	User
I@LOCK	JSH
I@LOGRG	Station
I@MAXDSZ	DM
I@MAXLM	DM
I@MAXME	System
I@MAXMQL	Interactive
I@MAXNUT	User
I@MAXPAD	User
I@MAXWPT	Disk
I@MCLDLY	System
I@MEC	System
I@MECCT	System
I@MED	System
I@MEMPRI	User
I@MERI	System
I@MESTOP	System
I@METL	System
I@METO	System

Operating System Parameter	Function
I@MEUCT	System
I@MIJID	System
I@MIJML	System
I@MIJPA	System
I@MIM	Account
I@MINPAD	User
I@MINWPT	Disk
I@MMEP	User
I@MMIN	User
I@MMIS	User
I@MP1SZ	System
I@MP2SZ	System
I@MP3SZ	Tape
I@MP4SC	Interactive
I@MP5SC	JSH
I@MP6SC	FSS
I@MPBS	System
I@MRD	Account
I@MSPLIT	JSH
I@MXCRDT	SL/MVS
I@MXM	Account
I@NCLASS	JSH
I@NCSP	System
I@NDSKBF	Station
I@NELOR	Disk
I@NEPHR	Disk
I@NERQT	Disk
I@NETDT	Tape
I@NETLF	Station
I@NEULFT	User
I@NGRN	System
I@NMI	JSH
I@NORRN	User
I@NOTEXT	Security
I@NPD	DM
I@NQLIM	System
I@NSCBFC	Station
I@NSLTRQ	SL/MVS
I@NSTAT	Station
I@NSTCH	Station
I@NSTCHO	Station
I@NSUBJ	Startup
I@NTXT	User

Table A-1. Operating System Parameters - Alphabetic (continued)

Operating System Parameter	Function
I@OEGNE	SL/MVS
I@OFFRNM	Station
I@OIPBMX	User
I@OPC	Account
I@ORI	User
I@PAUSE	User
I@PDDSK	Disk
I@PDMBUF	DM
I@PDOWN	DM
I@PDPAM	DM
I@PDPRIV	DM
I@PDRT	DM
I@PDSBFL	System
I@PDSLI	System
I@PDSSD	Disk
I@PERM	Disk
I@PFA	Account
I@PFS	Account
I@PPNDLY	Station
I@PRSZW	System
I@PWD	Account
I@PWWARN	RDM
I@RDJLWT	RDM
I@RDKEY	RDM
I@RNDRBN	Disk
I@RNGABT	Tape
I@RRJ	JSH
I@RRN	User
I@STAT	User
I@SBDECR	Station
I@SBIDLE	Station
I@SBINCR	Station
I@SBINIT	Station
I@SBU	User
I@SCPINS	Station
I@SCPTCR	Station
I@SDR	Startup
I@SEMACT	Account
I@SFEDI	Tape
I@SFEMI	Tape
I@SFEN	User
I@SGMAX	Station

Operating System Parameter	Function
I@SIMBFZ	System
I@SIMDSZ	System
I@SITNB	Station
I@SITNRQ	Station
I@SITNSQ	Station
I@SITRCE	Station
I@SLBFSZ	SL/MVS
I@SLBPAT	SL/MVS
I@SLDPBC	SL/MVS
I@SLFTMO	SL/MVS
I@SLITIM	SL/MVS
I@SLLIB	SL/MVS
I@SLNBUF	SL/MVS
I@SLNLP	SLT
I@SLNTMO	SL/MVS
I@SLOTIM	SL/MVS
I@SLT	SL/MVS
I@SLTBMX	SL/MVS
I@SLTKI	SL/MVS
I@SLTOWN	SL/MVS
I@SLUPI	SL/MVS
I@SLVL	Security
I@SLVLSG	Security
I@SMI	Account
I@SPMDLY	SPM
I@SPMMIN	SPM
I@SPMON	SPM
I@SPMTYP	SPM
I@SSDHPR	Disk
I@SSDMXT	Disk
I@STCHO	Station
I@STGERT	Station
I@STIN	User
I@STIS	User
I@STLN	Station
I@STRTHR	Station
I@STSMIN	User
I@STYPE	Disk
I@SYOWN1	DM
I@SYOWN2	DM
I@TBM	Account
I@TCHK	SL/MVS

Table A-1. Operating System Parameters - Alphabetic (continued)

Operating System Parameter	Function	Operating System Parameter	Function
I@TDBS	Tape	I@TSD	Account
I@TEXT	SL/MVS	I@TSM	Account
I@TFS	Account	I@TSMIN	JSH
I@THRESH	System	I@TSW	Account
I@TLMXBF	SL/MVS	I@TSX	Account
I@TMBS	Tape	I@TVM	Account
I@TMV	Tape	I@TWJ	Account
I@TNTB	Tape	I@UC	System
I@TODEF	System	I@USRSPM	SPM
I@TOMIN	System	I@UTSMIN	System
I@TRBGN	System	I@XMI	Account
I@TRNUM	System	I@ZJCLN	Startup
I@TS	Tape	I@ZOPT	JSH
I@TSBU	Account	NE@SDT	System

The following subsections describe the operating system parameters in the following order:

- Account
- Job Scheduler (JSH)
- Dataset Management (DM)
- Startup
- System
- Security
- User
- Disk
- Tape
- Station
- System Performance Monitor (SPM)
- Interactive
- FSS preemption (FSS)
- SUPERLINK Transport Task (SL/MVS)
- RDM

A.1.3.1 Account parameters

The account parameters are as follows (see subsection 5.3, The Installation-defined Accounting Algorithm, for examples of weighing factors):

Parameter	Type	Default	Significance
I@ACT	CD	1	Flag; if set equal to 1, accounting is mandatory. If set equal to 0, accounting is optional.
I@BRF	SV	0	Weighting factor used by accounting in determining the system billing units (SBUs) for blocks received from the front end.
I@BSF	SV	0	Weighting factor used by accounting in determining the SBUs for blocks sent to the front end.
I@CLC	SV	0	Weighting factor used by accounting in determining the SBUs for CLOSE calls made.
I@DMI	SV	0	Weighting factor used by accounting in determining the SBUs for the I/O wait time memory integral.
I@IOB	SV	0	Weighting factor used by accounting in determining the SBUs for disk sectors moved.
I@IOF	SV	0	Weighting factor used by accounting in determining the SBUs for FSS sectors moved. The allowable range is 0.00 through 5.00.
I@IOR	SV	0	Weighting factor used by accounting in determining the SBUs for user I/O requests.
I@MIM	SV	0	Weighting factor used by accounting in determining the SBUs for the minimum amount of memory used.
I@MRD	SV	0	Weighting factor used by accounting in determining the SBUs for memory-resident datasets.
I@MXM	SV	0	Weighting factor used by accounting in determining the SBUs for maximum memory used.

Parameter	Type	Default	Significance
I@OPC	SV	0	Weighting factor used by accounting in determining the SBUs for OPEN calls.
I@PFA	SV	0	Weighting factor used by accounting in determining the SBUs for permanent dataset space accessed.
I@PFS	SV	0	Weighting factor used by accounting in determining the SBUs for permanent dataset space saved.
I@PWD	CD	0	Flag; if set equal to 1, account password is mandatory.
I@SEMACT	CD	1	Flag; if set equal to 0, include time waiting semaphore with time spent executing. If not 0, separate time waiting semaphore from time executing.
I@SMI	SV	0	Weighing factor used by accounting in determining the system billing units for the semaphore wait memory integral.
I@TBM	SV	0	Weighting factor used by accounting in determining the SBUs for tape blocks moved.
I@TFS	SV	0	Weighting factor used by accounting in determining the SBUs for temporary file space used.
I@TSBU	SV	0	Minimum total of SBUs allowed for accounting.
I@TSD	SV	0	Weighting factor used by accounting in determining SBUs for time waiting for I/O.
I@TSM	SV	0	Weighting factor used by accounting in determining the SBUs for tape sectors (512 words) moved.
I@TSW	SV	0	Weighting factor used by accounting in determining SBUs for the time waiting to execute.
I@TSWS	SV	0	Weighting factor used by accounting to determine the system billing units of time waiting for semaphore.

Parameter	Type	Default	Significance
I@TSX	SV	0	Weighting factor used by accounting in determining SBUs for time executing in CPU.
I@TVM	SV	0	Weighting factor used by accounting in determining the SBUs for tape volumes mounted.
I@TWJ	SV	0	Weighting factor used by accounting in determining SBUs for time waiting for JXT.
I@XMI	SV	0	Weighting factor used by accounting in determining SBUs for the execution time memory integral.

A.1.3.2 Job Scheduler (JSH) parameters

Parameter	Type	Default	Significance
I@CSDMAX	CD	512	Maximum job class structure definition size, in words (multiples of 512). The corresponding JCSDEF parameter must match.
I@EXPANS	SV	50000_8	When a job is waiting for memory, the JSH does not make an allocation unless I@EXPANS words of memory remain free after the allocation.
I@JCCHAR	CD	16	Maximum job class characteristic size, in words. The corresponding JCSDEF parameter must match.
I@JNEMLM	CD	17400_8	Maximum number of blocks JSH allocates to a job not in Extended Memory Address (EMA) mode.
I@JOBMIN	SV	2	Minimum number of jobs in memory before expansion space is required.
I@JSCOS	CD	25	I@JSCOS is the disconnect cost in microseconds. When a job is disconnected from the CPU, I@JSCOS is subtracted from the job's time slice.
I@JSHTLE	SV	3	Number of seconds to extend the time limit after a job first encounters a time limit to enable termination processing.

Parameter	Type	Default	Significance
I@JSITS	CD	.1	When a job is granted a memory allocation, its first time slice is the value I@JSITS in seconds.
I@JSLK1	CD	2	I@JSLK1 is the in-memory thrash lock. After a job is placed in memory, the job is not eligible to roll out until at least I@JSLK1 seconds have elapsed. See subsection 4.3.2, Adjusting Memory Scheduling, for the range of allowable values.
I@JSLK2	CD	4	I@JSLK2 is the out-of-memory thrash lock. After a job is rolled out, the job is not eligible for a memory allocation until at least I@JSLK2 seconds have elapsed. See subsection 4.3.2, Adjusting Memory Scheduling, for the range of allowable values.
I@JSLK3	CD	0	Out-of-memory thrash lock multiplier. Any nonzero value makes the thrash lock proportional to the size of the job. A value of 1000 makes the lock 1 second for each second of roll-out time. See subsection 4.3.2, Adjusting Memory Scheduling, for the range of allowable values.
I@JSLK4	CD	0	In-memory thrash lock multiplier. See subsection 4.3.2, Adjusting Memory Scheduling, for the range of allowable values. Any nonzero value makes the thrash lock proportional to the size of the job. A value of 1000 makes the lock 1 second for each second of roll-in time.
I@JSMPA	CD	1.0	Coefficient used in the memory priority formula in priority units (see subsection 4.2)
I@JSMPB	CD	0.0	Coefficient used in the memory priority formula in words (see subsection 4.2)
I@JSMPC	CD	10	Coefficient used in the memory priority formula in seconds (see subsection 4.3).
I@JSMPD	CD	0.1	Coefficient used in the memory priority formula in priority units (see subsection 4.3).

Parameter	Type	Default	Significance
I@JSRRI	CD	20	If a job encounters a disk error during rollout, the roll is retried in I@JSRRI seconds.
I@JSTEI	CD	1	Jobs entering a delay state or an event wait state in response to a J$AWAIT or J$DELAY function are placed in a queue and checked every I@JSTEI seconds for completion of the event or the delay interval.
I@JSTS0	CD	.3	Coefficient used in the formula for calculating new time slice in seconds.
I@JSTS1	CD	.1	Coefficient used in the formula for calculating new time slice in seconds.
I@JSTS2	CD	0	Coefficient used in the formula for calculating new time slice in seconds.
I@JSTS3	CD	0	Coefficient used in the formula for calculating new time slice in seconds.
I@JTLDEF	CD	8	Default value for the job time limit in seconds.
I@JXTSIZ	CD	15	Defines the number of Job Execution Table (JXT) entries; the maximum number of jobs that can be active. The allowable range is 1 through 255
I@LOCK	CD	0	Job lockout status when Recovery of Rolled Jobs (RRJ) detects the system being started is not the same system in effect when the job was rolled out. This parameter is for the use of jobs having JTEPC nonzero. The values are: 0 Recover normally 1 Recover and lock job out 2 Make job not recoverable
I@MP5SZ	CD	I@NTXT*12	Size, in words, of QUEPOOL for JSH, as a function of I@NTXT
I@MSPLIT	CD	2000_8	Minimum dataset size to stripe a roll image

Parameter	Type	Default	Significance
I@NCLASS	CD	32	Maximum number of job classes that can be defined. The corresponding JCSDEF parameter must match.
I@NMI	CD	1	Type of memory integral used. For old memory integrals, set I@NMI to 0; for new memory integrals, set I@NMI to 1. The old memory integrals charge the user for memory usage on a per task basis. The new memory integrals charge the user on a per job basis.
I@RRJ	CD	1	Recovery of Rolled Jobs (RRJ). This parameter checks for 0 for recovery to be disabled; any nonzero value enables recovery.
I@SEMPEN	CD	2.0	Coefficient for the semaphore wait penalty factor used in the formula for computing a new time slice.
I@TSMIN	SV	3	Before the scheduler requests EXEC to connect a job to the CPU, JSH ensures that the job to be connected has a time slice of at least I@TSMIN (in milliseconds). The minimum value must be larger than 1 millisecond.
I@ZOPT	CD	2	System startup option default. The parameter values and their descriptions are as follows: 0 Install 1 Deadstart 2 Restart

A.1.3.3 Dataset Management (DM) parameters

Parameter	Type	Default	Significance
I@DEFLM	CD	100000	Default dataset size limit, in sectors.
I@DTDREP	CD	0	Defines whether a second reporting of disk write errors is sent to DQM after all data is on disk. If the second reply is sent, a performance penalty is paid. The following parameter values and their descriptions follow: 0 One reply is send 1 A second reply is sent after all data is on disk

Parameter	Type	Default	Significance
I@DXTFUL	CD	80	Percentage of DXT that must be full to have Startup issue a warning message. The default indicates that the DXT must be at least 80% full before Startup will issue the message.
I@DXTLDV	CD	0	Device that the DXT starts on; if 0, DXT starts on the master device. If a valid logical device name is specified, Startup attempts to put the DXT on that device. If DD-XX-XX is specified, the DXT can go on any device.
I@DXTOVF	CD	'N'R	DXT does not overflow to a second device
I@DXTSZ	CD	I@NPD/8	DXT size in 512-word blocks
I@FDPMAX			The number of translation datasets that can be operated on simultaneously.
I@MAXDSZ	SV	I@MAXLM*512	Maximum dataset size in words
I@MAXLM	CD	200000	Maximum dataset size limit in sectors
I@NPD	CD	1000	Maximum number of permanent datasets
I@PDOWN	SV	'USN'L	Defines the identifier in dataset ownership. The parameter values and their descriptions are as follows: 'ACN'L Account number is the owner 'USN'L User number is the owner
I@PDMBUF	CD	4	Number of PDM I/O buffers (sectors) in PDM page table, 'PDMTGT'.
I@PDPAM	CD	200_8	The default public access mode for permanent datasets. The parameter values and their descriptions follow: 011_8 Execute only permission 001_8 Read permission 002_8 Write only 004_8 Maintenance permission 200_8 No public access
I@PDPRIV	CD	1	Permanent dataset privacy. The parameters and their descriptions follow: 0 Disabled 1 Enabled

Parameter	Type	Default	Significance
I@PDRT	CD	1	Default permanent dataset retention period in days
I@SYOWN1	SV	A'SYSTEM'L	System permanent dataset ownership value (characters 1-8)
I@SYOWN2	SV	0	System permanent dataset ownership value (characters 9-15)
I@WPDS	CD	10	Number of retries after the 5-second delay for permanent dataset (PDS) full condition for jobs. If 0, it is treated as infinite. If count overflows, the job is aborted.

A.1.3.4 Startup parameters

Parameter	Type	Default	Significance
I@NSUBJ	CD	64	Maximum number of *SUBMIT directives in the Startup parameter file.
I@SDR	CD	0	Enable SDR recovery. Startup does not examine this parameter.
I@ZJCLN	SV	'STARTUP'L	Startup job class name.

A.1.3.5 System parameters

Parameter	Type	Default	Significance
I@BFDECR	SV	I@SBDECR	System Buffer decrement in words. This Buffer Memory request decrement is the amount of which to decrease buffer size.
I@BFIDLE	SV	I@SBIDLE	System Buffer idle size in words
I@BFINCR	SV	I@SBINCR	System Buffer Memory request increment in words
I@BFSIZE	SV	20000_8	Initial system buffer size in words
I@BRBGN	CD	01_8	First B register to save
I@BRNUM	CD	17_8	Number of B registers to save
I@CECT	CD	0	Number of correctable errors to allow on a chip before correction is attempted

Parameter	Type	Default	Significance
I@CEDDI	CD	12_8	Correctable (single bit) memory error disable detection interval in minutes. This is the interval detection that will be disabled when I@MECCT correctable errors have been encountered.
I@CHATIM	CD	5	User driver time-out default in tenths of a second
I@CMETL	CD	10	Number of entries in the correctable memory error reporting table
I@COSMIN	CD	1	Minimum percent of CPU in COS when COS and GOS are busy
I@DDCCH	CD	4	The default channel number for the DDC, the Deadstart Dump routine on system without an IOS
I@DDCUN	CD	0	The default unit for the DDC. (The Deadstart Dump routine on systems without an IOS.)
I@DEFINT	CD	1	History trace heartbeat in seconds (each CPU)
I@DMPSIZ	CD	10117000_8	Number of words to reserve for system dump. Two additional sectors are reserved for dump header.
I@DVLRES	CD	2	Number of tracks reserved for writing device labels.
I@FIT	SD	0	This parameter controls the checking of front-end connections to the Cray mainframe. If I@FIT is zero, then ID checking is not done. If I@FIT=FIT@ID then only IDs are checked. If I@FIT=FIT@ADD then IDs and remote addresses are checked. This allows COS to distinguish between a reliable station and a perpetrator. If this parameter is

Parameter	Type	Default	Significance
I@FIT (continued)			nonzero, then the front-end connections can be configured ON or OFF either in the restart file or by the master operator after startup.
I@GOSPWS	CD	80* C@CPQUAN+16	Size of buffer reserved in EXEC for the GOS PWS.
I@GOSSIZ	CD	$4,000,000_8$	Default size of GOS (1 Mword)
I@GOUMIN	CD	100,000	Minimum number of words to be allocated for GOS users before GOS will be allowed to start.
I@INQLIM	SV	NE@SDT/2	Maximum input queue size in SDT entries. This data structure is 112_8 words in length.
I@IOPICH	CD	C@CPMCHN	IOP-0 to CPU communication channel
I@LGBSZ	CD	2000_8	$SYSLOG buffer size in words
I@LGDSZ	CD	2000	$SYSLOG dataset size in sectors
I@MCLDLY	CD	1000	Number of microseconds to wait for input during link master clear sequence (master clear delay time)
I@MEC	CD	1	If nonzero, correction is attempted on single-bit errors
I@MECCT	CD	200	Number of correctable memory errors allowed before disabling single-bit error reporting
I@MED	CD	1	If nonzero, correctable memory error detection is enabled
I@MERI	CD	2	Memory error reporting reset interval in minutes. This interval is used to flush the correctable memory error reporting table.
I@MESTOP	CD	1	If 1, stop if error in idle

Parameter	Type	Default	Significance
I@METO	CD	2	Time interval, in seconds, that measures maximum number of memory errors allowed (see also I@MECCT and I@MEUCT)
I@MEUCT	CD	20	Number of uncorrectable (double-bit) errors allowed in noncritical areas before stopping the system
I@MIJID	CD	20	Maximum number of interjob transfer IDs
I@MIJML	CD	1000_8	Maximum size, in words, of an interjob message
I@MIJPA	CD	50	Maximum number of interjob communication paths
I@MP1SZ	CD	2000	Size, in words, of memory pool 1. Must be at least as large as 1 MCU segment plus 2 words. SCP uses this space for communication with the MCU until Startup completes and the system buffer is built.
I@MP2SZ	CD	512	Size of memory pool 2. one sector for memory error logging.
I@MPBS	CD	15	Maximum number of linked interjob parameter blocks
I@NCSP	CD	1	Number of copies of CSP on disk. The parameter values and their descriptions follow: 0 CSP remains memory resident and is not written to disk. ≥1 CSP is not memory resident following startup. If the value is greater than 1, copies reside on different devices and channels. Multiple copies of CSP allow multiple jobs to access a disk copy of CSP without device or channel conflicts.
I@NGRN	CD	1	Number of generic resources to be declared with GRT-built directive in STPTAB

Parameter	Type	Default	Significance
I@PDSBFL	CD	512*18	PDSDUMP/PDSLOAD buffer size in words. This value must be a multiple of 512.
I@PRSZW	CD	60_8	Octal percentage of priority in output queue for size waiting factor
I@PSDLI	Cd	000300_8	Pseudo deadlock interrupt
I@SIMBFZ	SZ	1	Default buffer size, in sectors, if on simulator
I@SIMDSZ	CD	4000_8	Number of words to reserve for system dump when executing in simulator. This includes space for future inclusion of BIT/V registers.
I@THRESH	SV	25	Used in intertask communications to determine if a called higher priority task has been inactive long enough
I@TODEF	CD	500	Event recall time-out default in msecs
I@TOMIN	CD	10	Event recall time-out minimum in msecs
I@TRBGN	CD	0	First T register to save
I@TRNUM	CD	20_8	Number of T registers to save
I@UC	CD	0	Flag; if I@UC≠0, user channels are in effect.
I@UTSMIN	SV	500	Minimum user-requested time slice, in microseconds, for F$SPY processing. The parameter values and their descriptions follow: 1000 3% system overhead (1/S) 100 23% system overhead (1/S) 15 100% system overhead, system dies
NE@SDT	CD	200	Number of entries in SDT

A.1.3.6 Security parameters

Parameter	Type	Default	Significance
I@CRYPT	CD	0	Password encryption. If 1, encrypt all passwords.

Parameter	Type	Default	Significance
I@NOTEXT	CD	-1	Flag that allows the suppression of text. The parameter descriptions and their values follow: -1 Text is suppressed 0 Text is displayed
I@SLVL	CD	1	System security level. The parameter values and their descriptions follow: -1 Ignore all security checks 0 Issue warning messages only 1 Implement full system security
I@SLVLSG	CD	-1	Security level of $SYSTEMLOG. The parameter values and their descriptions follow: -1 Allow unedited statement echoes 0 Do not allow unedited statement echoes.

A.1.3.7 User parameters

Parameter	Type	Default	Significance
I@AUTOFL	CD	1	Flag; if set equal to 1, enables automatic user field length reduction.
I@AVL	CV	0	Additional vector logical unit is disabled if 0. This parameter should be set to 0 if C@CPAVL=0.
I@BDM	CD	0	Disable bidirectional memory. The parameter values and their descriptions follow: 0 Bidirectional memory transfer is off by default; user can enable. 1 Bidirectional memory transfer is on by default, user can disable. This parameter must be 0 for CRAY-1 computer systems.
I@BFI	CD, SD	033_8	ASCII character to be used as default blank field initiator; 777_8 indicates no blank field compression.
I@BULLIT	CD	0	System installation dataset option. The parameter values and their descriptions follow: 0 No bulletin dataset 1 Bulletin dataset

Parameter	Type	Default	Significance
I@DNBFZ	SV	8	Default CIO buffer size, in sectors, for datasets
I@DNDT1		'*'R	Preferred device type 1
I@DNDT2		0	Preferred device type 2
I@DNDT3		0	Preferred device type 3
I@DNOVF		0	Flag that sets the dataset overflow condition. The parameter values and there definitions follow: 0 Datasets are allowed to overflow devices 1 Datasets are not allowed to overflow devices
I@DNPDMO		0	Sets default buffer flush for PDM requests, as follows: 0 Flush 1 Close
I@DNSPD		0	Default user striping size
I@DNSZ		0	Default dataset size
I@DNXMAX		0	Maximum transfer request size. If this value is zero, values will be half the buffer size for blocked datasets. This value must be 0 for unblocked or random datasets.
I@DNXMIN		0	Minimum transfer request size. If this value is zero, then the values will be half the buffer size for blocked datasets. This value must be the buffer size.
I@DPWAIT	SD,CD	0	Default value for DISPOSE=WAIT. The parameter values and their descriptions follow: 0 NOWAIT 1 WAIT
I@EFI	CD	1	Enable floating-point interrupt detection. The parameter values and their descriptions follow: 0 Off by default, user must enable 1 On by default, user must disable

Parameter	Type	Default	Significance
I@EMA	CD	0	Extended Memory Addressing (EMA). If you want to use the EMA modules, set this parameter to 1. C@CPEMA must be set to 1 if I@EMA is set to 1. I@EMA must be 1 on a CRAY Y-MP and CRAY X-MP EA machine.
I@FEMSK	SD	-1	Bit mask allowing user control over fatal errors. The parameter values and their descriptions follow: 0 Error is fatal for job -1 Error is fatal for job step
I@IJTL	CD	10000_8	Initial length, in words, of the Job Table Area (JTA). This parameter must be a multiple of 1000_8 and large enough to accommodate the fixed portion of the JTA, plus a minimum of 3 DNTs.
I@JFLDEF	CD	I@JFLMAX	Default maximum job field length in blocks (1 block = 512 words)
I@JFLMAX	CD	C@MMSIZE/ 1000_8 - 1100^8 - I@IJTL/ 1000_8 - I@SBINIT/ 1000_8	Maximum amount of memory, in blocks, that a job can use (excluding the JTA)
I@JFLMSG	SV	0	Flag; disables field length change and OPEN/CLOSE messages, if 0.
I@LGUSZ	CD	2000_8	$LOG size limit in sectors
I@MAXNUT	CD	1	Maximum number of user tasks allowed per user job. I@MAXNUT=1 disables multitasking.
I@MAXPAD	CD	10000_8	When a user job relinquishes memory, the memory is returned to the system if the total amount of unused space in the job exceeds I@MAXPAD words.
I@MEMPRI	CD	7	Default priority value when no P parameter is given on JOB statement

Parameter	Type	Default	Significance
I@MINPAD	CD	4000_8	Minimum increase, in words, of memory guaranteed by the system when a job requests additional memory
I@MMEP	CD	2	Managed memory epsilon, in words, for heaps. This parameter must be at least LE@HP.
I@MMIN	CD	4000_8	Default size, in words, of increments to run-time managed memory
I@MMIS	CD	I@STIS+340_8	Initial size for run-time managed memory in words
I@NEULFT	CD	20	Number of LFT entries in each user-area LFT block (cannot exceed 64)
I@NORRN	CD	0	No Rerun Checking flag. The parameter values and their descriptions follow: 0 Enable no rerun checking ≠0 Disable no rerun checking
I@NTXT	CD	I@JXTSIZ	Defines the number of Task Execution Table (TXT) entries; the maximum number of user tasks that can be active.
I@OIPBMX	CD	32	Maximum length, in words, of the IPC send and receive parameter blocks. Must not be less than 32 decimal words.
I@ORI	CD	0	Disable operand range error detection. The parameter values and their descriptions follow: 0 Operand range error detection is off by default 1 Operand range error detection is on by default This parameter must be 0 on CRAY-1 computer systems.
I@PAUSE	CD	0	Option for semantic meaning of PAUSE statement in CFT. The parameter values and their descriptions follow: 0 Suspends CFT program until operator intervention 1 Terminates CFT program

Parameter	Type	Default	Significance
I@RRN	CD	0	Job Rerun flag. The parameter values and their descriptions follow: 0 Enable job rerun Nonzero Disable job rerun
I@SBU	CD	0	Flag; if set, specifies that the SBU totals are to be printed in the user logfile.
I@STAT	CD	0	Flag that determines if dataset statistics are enabled or disabled. The parameter values and their descriptions follow: 1 Enable dataset statistics 0 Disable dataset statistics Default overridden by OPTION,STAT.
I@SFEN	CD	1	Flag that determines if front-end servicing is enabled. The parameter values and their descriptions follow: 0 Disable 1 Enable
I@STIN	CD	400_8	Default size, in words, for memory-managed stack increments
I@STIS	CD	4000_8	Initial stack size, in words, for run-time memory manager
I@STSMIN	SV	50	Minimum user requested time slice, in microseconds, for F$PROF processing
I@$INSZ	SV	4	Default CIO buffer size, in sectors, for job's $IN dataset
I@$ODLM	CD	2000	Default $OUT size limit in sectors
I@$OMLM	CD	20000_8	Maximum $OUT size limit in sectors

A.1.3.8 Disk parameters

Parameter	Type	Default	Significance
I@CPAGEI	CD	2000	Number of milliseconds between reevaluations of control path activity
I@CPDSK	CD	20	Number of milliseconds to add to window control path active time for selecting a disk device

Parameter	Type	Default	Significance
I@CPFSS	CD	1	Number of milliseconds to add to window control path active time for selecting a SSD solid state storage device and BMR
I@DD	CD	1	Total number of disk units configured in the Equipment Table (EQT)
I@DXTCAI	CD	'N'R	Contiguous allocation indexes (AIs) required for DXT. The parameter values and their descriptions follow: 'N'R No contiguous AIs 'Y'R Contiguous AIs
I@MAXWPT	CD	512*48	Maximum track size, in words, for all disks. This value should be the maximum size of any of the categories below that fits the system being configured: 512*42 if DD-49s 512*24 if DD-39s 512*18 if DD-29s 512*32 if an SSD 512*48 if DD-40s
I@MINWPT	CD	512*18	Minimum track size, in words, for all disks
I@NELOR	CD	64	Number of LOR entries
I@NEPHR	CD	512	Number of PHR entries on the free queue
I@NERQT	CD	256	Number of RQT entries on the free queue
I@PERM	CD	0	Flag that disallows permanent dataset on scratch device. The parameter values and their description follow: 0 disallow permanent dataset 1 allow permanent dataset
I@RNDRBN	CD	0	Allows a site to have the system select a new device based on either control path activity or by round-robin access to the control paths. A value of 1 invokes the round-robin means and allows a site to balance its channel (control path) activity and minimize system I/O wait

Parameter	Type	Default	Significance
I@RNDRBN (continued)			time. A value of 0 causes the selection of a new device for allocation on the control path with the least activity.
I@SSDHPR	CD	C@CPQUAN	Defines the number of SSD Hot Path requests that may be simultaneously outstanding, either in the Hot Path Request Queue or currently being processed. Equating this parameter to the number of CPUs may simultaneously execute a synchronous SSD Hot Path Request. It may be desirable to increase this value if the amount of asynchronous SSD Hot Path I/O activity is keeping the the Hot Path Request Queue full.
I@SSDMXT	CD	D'60*C@NSV HSP*C@SSD MTR/@TRSD8	Maximum transfer size, in sectors, permitted on the SSD Hot Path. Larger transfers will use the DQM Path.
I@STYPE	CD	0	Determines if unassigned dataset is allocated on permanent space device. The parameter values and their descriptions follow: 0 Allocated to permanent space device 1 Allocated to scratch device

A.1.3.9 Tape parameters

Parameter	Type	Default	Significance
I@NETDT	CD	0	Number of IOS tape drives configured in the Tape Device Table (TDT)
I@SFEDI	CD	0	Default servicing front-end ID
I@SFEMI	CD	0	Mandatory servicing front-end ID

Parameter	Type	Default	Significance
I@TNTB	CD	1	Number of tape banks in the system. TQM attempts to allocate tape devices in different banks to avoid path conflicts. Tape banks are defined to be physically and logically separated from each other. This means a particular device, controller, or channel cannot be accessible in more than 1 bank.
I@TS	CD	0	Flag that enables tape I/Os. The parameter values and their descriptions follow: 0 Disable assembly of TQM and TDT 1 Enable assembly of TQM and TDT
I@TQMTRC	CD	1	Disables/enables TQM tracing, as follows: 0 Disable TQM tracing 1 Enable TQM tracing
I@TDTEXT		1	Allow write extension to scratch volumes. The parameters follow: O Do not allow extension 1 Extend
I@BPL	CD	1	Flag that allows by-passing label processing. The parameter values and their descriptions follow: 0 No by-passing is allowed 1 By-passing allowed for privileged users under a system with full security turned on; otherwise, all users are allowed.
I@DTDBS	CD	32768	Default tape block size in bytes
I@ML		1	Number of media loaders
I@MLHN		'CRAY'L	Host name for media loaders
I@MP3SZ	CD	I@NETDT*$$TABLE	Size of memory pool 3, used by TQM

Parameter	Type	Default	Significance
I@RNGABT	CD	1	Flag; if set, on-line tape ring processing aborts jobs attempting to write on tapes that were ACCESSed with RING=OUT specified. The operator messages on the servicing front-end console contains information about the write ring in the initial mount message, if set. If not set, ring processing is not performed.
I@TDBS	CD	'32768'L	Default tape block size in bytes, left-justified. I@TDBS is the default interchange maximum tape block size in bytes. I@TDBS is expressed as a left-justified, zero-filled ASCII character string. Leaving I@TDBS at its released value provides flexibility in handling a variety of tapes.
I@TMBS	CD	1048576	Maximum tape block size in bytes (default = 1,048,576). This should not be set to less than 32,768, or more than two-fifths the amount of Buffer Memory allocated to the XIOP. If the site does not use large block sizes, the recommended value is 131,072.
I@TMV	CD	255	Maximum number of volumes and reels that can comprise any tape dataset (255 maximum)

A.1.3.10 Station parameters

Parameter	Type	Default	Significance
I@ALLSDT	SV	1	Flag; if set equal to 1, enables the display of all SDT entries on all station displays.
I@ALLST	CD	1	Flag, if set to 1, allows stations to request job and link statuses for all stations.
I@APLN	SV	360_8	Length of AP trace table in bits. 0 disables tracing.

Parameter	Type	Default	Significance
I@DTRDLY	CD	4	Dataset transfer delay count. Number of messages to wait before reissuing a dataset transfer request.
I@GROUP	SV	0	Allows a group of stations to be viewed as a logical entity. The first letter of a station ID is its major letter. The second letter is its minor letter. The parameter values and their descriptions follow: 0 Allows a group of stations to be viewed as a logical entity 1 Enables group processing within SCP
I@IAPOLL	SV	500	Unbuffered interactive poll rate in msec
I@IMXTXT	CD	64	Maximum text field length in words (default is 512 characters)
I@NETLF	CD	0	Number of logging front-end systems
I@NSCBFC	CD	32	Number of NSC buffers. This parameter must be the same as the NSCBFC parameter.
I@INSMX	CD	I@INST	Maximum number of internal station connections allowed
I@INST	CD	0	Number of internal stations
I@INTERN	CD	1	Switch that enables or disables an internal station. The parameter values and there descriptions follow: 0 Internal station is disabled 1 Internal station is enabled
I@LOGRQ	CD	0	Flag that allows any station to write to another's $log dataset. The parameter values and their descriptions follow: 0 Disable Nonzero Enable
I@NDSKBF	SV	2	Default number of disk sectors used for disk buffer by STG for a stream

Parameter	Type	Default	Significance
I@NSTAT	CD	2	Maximum number of front-end stations logged on at one time. This determines the size of the Link Extension Table (LXT), which contains information associated with each logged on front-end ID. The interactive station and IOS must be counted as front-end stations.
I@NSTCH	CD	1	Number of Link Configuration Table (LCT) and Link Interface Table (LIT) entries. There is one entry for each Cray channel that is defined to be connected to a front-end computer.
I@NSTCHO	CD	9	Maximum IOP channel ordinal allocated for front-end communication through the IOS. This is the number of Channel Extension Table (CXT) entries. The CXT is used by the EXEC to communicate front-end parameters to MIOP. This parameter must be greater than or equal to the I@NCOR parameter, plus the maximum number of front-end identifiers allowed on all NSC and VME (FEI-3) channels. See the I@NNSC, I@NVME, I@NNID and I@NVID parameters. (This parameter is 0 if no IOS is present.)
I@OFFRNM	CD	'SCPOFFER'	The name that SCP uses to offer an IPC connection. This is an 8-character (maximum) string.
I@PPNDLY	SV	60	Time to delay, in seconds, for postpones
I@SBDECR	CD	100000_8	Decrement value, in words, used for releasing station buffers; should be at least twice I@SBINCR.
I@SBIDLE	CD	2000_8	System buffer idle size in words. If the space is smaller than the decrement threshold, but fewer than I@SBIDLE words are in use, the remaining space is returned to the system. This parameter must be less than I@SBINIT.

Parameter	Type	Default	Significance
I@SBINCR	CD	40000_8	Increment value, in words, used for requesting additional memory for station buffers. I@SBINCR should be at least the size of the largest segment and disk buffer size for a stream.
I@SBINIT	CD	20000_8	Initial size of memory, in words, to be used for station buffers
I@SCPINS	SV	1	Minimum amount of time, in seconds, between statuses of the internal station OFFER
I@SCPTCR	SV	1	Number of seconds between update of transfer rates for all active front-end dataset transfers
I@SGMAX	CD	14_8	Maximum segment length, in words, to trace in the SCP trace table
I@SITNBQ	CD	2	The number of buffer queue items for all SCP internal connection tables. This number must be at least 2.
I@SITNRQ	CD	4	The number of receive queue items for all SCP internal connection tables. This number must be at least 4.
I@SITNSQ	CD	2	The number of send queue items for all SCP internal connection tables. This number must be at least 2.
I@SITRCE	CD	10	The number of trace entries for SCP IPC trace table. Zero disables tracing.
I@STGERT	SD	2	Maximum PDM error retry count used by STG; must be greater than zero.
I@STLN	CD	360_8	Length, in words, of other station's trace table
I@STRTHR	SV	1240_8	Block transfer limit, in sectors, for streaming

A.1.3.11 System Performance Monitor (SPM) parameters

Parameter	Type	Default	Significance
I@SPMDLY	SV	1800	SPM collection interval in seconds

Parameter	Type	Default	Significance
I@SPMMIN	SV	10	Delay, in seconds, when SPM needs memory
I@SPMON	SV	1	SPM Task Enable flag. The parameter values and their descriptions follow: 0 Disable SPM 1 Enable SPM
I@SPMTYP	SV	7777_8	SPM subtype enable vector. Each bit enables the collection of data for a corresponding subtype if set. Bit 63 corresponds to subtype 12 and bit 52 to subtype 1.
I@USRSPM	CD	0	If nonzero, users can initialize SPM task using the F$SPM call

A.1.3.12 Interactive parameters

Parameter	Type	Default	Significance
I@IAAUT	CD	1	Number of Active User Tables allocated (maximum number of interactive users logged on). This is the same as the maximum interactive process number. The process number is used to differentiate between interactive users; see the Front-end Protocol Internal Reference Manual. This parameter should be defined to allow at least one IOS interactive terminal at IOP sites. (I@IAAUT = IA$MAXPN, an IOS parameter.)
I@IAIBT	CD	3	Number of words in the bit map for interactive buffers with each bit representing a 10-word (80-character) buffer (1-bit map word represents 640 words). The configuration used by Software Development allocates 1 word per interactive user (I@IAIBT = I@IAAUT). This should be more than sufficient for normal use.
I@IACTR	CD	4	Number of account processing retries allowed for an interactive job

Parameter	Type	Default	Significance
I@MAXMQL	CD	20	Maximum number of interactive output lines for a single user. When the user job attempts to queue more lines than the value of this parameter allows, the job is suspended until the station receives a line. This value must be nonzero for support of operator messages.

A.1.3.13 FSS preemption parameter

Parameter	Type	Default	Significance
I@MP6SZ	CD	0	Determines the size of the memory pool, in words, used to house tables that regulate the FSS preemption processes. If preemption is desired, this parameter must be at least 512 words.

A.1.3.14 SUPERLINK/MVS (SL/MVS) parameters

Parameter	Type	Default	Significance
I@MXCRDT	CD	17	Controls allocation of credit to peer transport entity. Should not be adjusted.
I@NSLTRQ	CD	10	Maximum number of intertask requests queued to SLT
I@OEGNE	CD	256	Number of activities allowed in the SUPERLINK transport service. If SUPERLINK is disabled, change this to 0.
I@SLBFSZ	CD	8232	Default FEI link buffer size in words. This parameter must not be less than 8192 words.
I@SLBPAT	CD	0111111111106666644444_8	Diagnostic bit pattern used in all FEI protocol messages.
I@SLDPBC	CD	5	Maximum number of messages sent on a half-duplex link before the link must change to input.

Parameter	Type	Default	Significance
I@SLFTMO	CD	30	FEI driver time-out, in seconds, for read and write operations in tenths of a second.
I@SLITIM	CD	10000	Maximum time interval, in milliseconds, during which a half-duplex FEI link may remain idle without changing from output to input.
I@SLLIB	CD	0	Flag that specifies if SUPERLINK Library support is assembled. The parameter values and their descriptions follow: 0 Do not assemble SUPERLINK Library support 1 Assemble SUPERLINK Library support
I@SLNBUF	CD	1	Number of buffers of data the NSC HYPERchannel driver can read without a read request from SLT
I@SLNLP	CD	2	Logical path ID for SLT as a user of the NSC HYPERchannel.
I@SLNTMO	CD	30	NSC driver time-out for read and write operations in tenths of a second
I@SLOTIM	CD	100	Throttle delay for data output on FEI links, in milliseconds
I@SLT	CD	1	Flag that enables or disables assembly of SLT. The parameter values and their descriptions follow: 0 Disable assembly of SLT 1 Enable assembly of SLT
I@SLTBMX	CD	400000_8	Maximum number of words of SYSBUF SLT can use at one time. Must contain an allowance of I@SLBFSZ words for every link configured, plus store for SLT activities (approximately 200 words per transport connection) plus store for normal operational data traffic. When SLT is close to its maximum store usage, performance may be seriously degraded.

Parameter	Type	Default	Significance
I@SLTKI	CD	300	Token word time, in milliseconds, on half-duplex FEI links.
I@SLTOWN	CD	1	SLTs reserved value for the OWN field in NSC HYPERchannel driver N-packets
I@SLUPI	CD	300	Idle message interval, in milliseconds, on FEI links
I@TCHK	CD	0	Flag that specifies whether or not to use checksums in transport protocol messages. The parameter values and their descriptions follow: 0 Do not use 1 Use
I@TEXT	CD	0	Flag that specifies whether to use normal or extended format transport messages. The parameter values and their descriptions follow: 0 Normal 1 Extended
I@TLMXBF	CD	15	Maximum transport credit to accept on a connection

A.1.3.15 RDM user limit parameters

Parameter	Type	Default	Significance
I@DLPP	CD	60	Default user LPP value
I@DPAM	CD	1	Default user PAM value
I@PWWARN	CD	30	RDM warning that password expires in 30 days
I@RDJLWT	CD	0	RDM job limit wait parameter. The possible values and their descriptions follow: 0 Wait and retry 1 Abort the user of over the job limit
I@RDKEY	CD	1	User RD entry key for PDM functions. The following is a list of values and their descriptions: 0 Account number 1 User number

A.2 I/O SUBSYSTEM INSTALLATION PARAMETERS

See the I/O Subsystem (IOS) Administrator's Guide, publication SG-0307, for complete information on the IOS installation parameters.

B. UPDATE PROGRAM LIBRARIES

Decks in the following program libraries constitute CRI software running on the CRAY Y-MP, CRAY X-MP EA, CRAY X-MP, and CRAY-1 computer systems. Decks modified and reassembled to generate a new binary for a program are listed in the left column. Routines affected by the change are listed in the right column.

B.1 ARLIBPL

All decks within ARLIBPL comprise $ARLIB. All routines in $ARLIB are written in CAL.

B.2 CALPL

Decks	Routines
C	APML, OLDCAL
BGNCAL2 --> ENDCAL2	CAL

B.3 CLIBPL

All decks within CLIBPL comprise $CLIB.

B.4 COSPL

Each COS task has a separate deck name.

Decks	Routines
CSP	CSP
CT	COSTXT/COSDEF
DDC	DDC
EXEC	EXEC
GLOBAL-->STARTUP	System tasks
UT	$UTLTXT/$UTLDEF

B.5 COSUTPL

Decks	Routines
ACCOUNT	ACCOUNT
ACCOUNT1	ACCOUNT1
ACCTDEF	ACCTDEF
ALTBCD	ALTBCD
AUDIT	AUDIT
BACKUP	BACKUP
BINDGOS	BINDGOS
BLOKGOS	BLOKGOS
BUPIO	BUPIO
BVCEDIT	BVCEDIT
CHARGES	CHARGES
CHNGGOS	CHNGGOS
CLEANUP	CLEANUP
CLUPIO	CLUPIO
DUMP	DUMP
DUMPGOS	DUMPGOS
EXTRACT	EXTRACT
FDUMP	FDUMP
GENBCD	GENBCD
GENMCD	GENMCD
INITT	INITT
JCSDEF	JCSDEF
LOADCAT	LOADCAT
LOADGOS	LOADGOS
MANAGE	MANAGE
MIGRATE	MIGRATE
PASSWRD	PASSWRD
PDMCAT	PDMCAT
PDSDUMP	PDSDUMP
PDSLOAD	PDSLOAD
PRVDEF	PRVDEF
RDACC	RDACC
RDAGET	RDAGET
RDAPUT	RDAPUT
RDAUDIT	RDAUDIT
RDEDIT	RDEDIT
RDGEN	RDGEN
RDMERGE	RDMERGE
RDMLIB1,RDMLIB2	$RDMLIB
RDNRD	RDNRD
RECALL	RECALL
RECIO	RECIO
RELOAD	RELOAD
RESTORE	RESTORE
RETIRE	RETIRE
ROUTE	ROUTE
SETOWN	SETOWN

Decks	Routines
SSAF	SSAF
STATGOS	STATGOS
STATMCD	STATMCD
STATS	STATS
STOPGOS	STOPGOS
TDI	TDI
TDUMP	TDUMP
TG	TG
VALBCD	VALBCD
VOLMAP	VOLMAP
ZZZADMON --->ZZZADMOF	$ADMLIB

B.6 CRAY1PL

Decks	Routines
AHT	AHT
ARB	ARB
ARM	ARM
BRB	BRB
CMD	CMD
MIT	MIT
SFA	SFA
SFM	SFM
SFR	SFR
SIS	SIS
SR3	SR3
SRA	SRA
SRB	SRB
SRL	SRL
SRS	SRS
STAN	STAN
SVC	SVC
TRB	TRB
VPOP	VPOP
VRA	VRA
VRL	VRL
VRN	VRN
VRR	VRR
VRS	VRS

B.7 CSIMPL

All decks within CSIMPL comprise CSIM. An EOF in the CSC deck separates the routines written in Fortran (first file) from those written in CAL (second file).

B.8 DBGPL

Decks	Routines
DBIMSG	$DBMSGS
DBMSGS	
DBALL-->DBALL*	
DBDS-->DBDS*	$UTLIB
DBSDB-->DBSDB*	
DBDD-->DBDD*	
DBDDD-->DBDDD*	DDA
DBGDDA-->DBGDDA*	
DBDD-->DBDD*	
DBDDD-->DBDDD*	DRD
DBGDRD-->DBGDRD*	
DBDD-->DBDD*	DEBUG
DBDDD-->DBDDD*	DRD

B.9 DIAGPL

Decks	Routines
BMXTAP	BMXTAP
CLEARIO	CLEARIO
DDTEST	DDTEST
DELAY	DELAY
DIAGLB	DIAGLB
DONUT	DONUT
DSDIAG,DSDIAGD	
DSMOS16K,DSIOM	
DSIOP,DSMOS,DSHSP	DSDIAG
DSLSP	
HERG	HERG
MENULIB	MENULIB

Decks	Routines
OLCFDT	OLCFDT
OLDMON	OLDMON
OLNET	OLNET

B.10 FDCPL

All decks within FDCPL are built into $UTLIB. Deck EOF1 separates the routines written in CAL (first file) from those written in Fortran (second file).

B.11 FLOWPL

All decks within in FLOWPL are built into $UTLIB. Deck EOF1 separates the routines written in CAL (first file) from those written in Fortran (second file).

B.12 FTLIBPL

All decks within FTLIBPL comprise $FTLIB. All routines in $FTLIB are written in CAL.

B.13 GENPL

Decks	Routines
DELTEMP	Temporary dataset cleanup job
GENARCH	Archiving utilities generation job
GENAUT	Autotasking utilities generation job
GENBSCC	Bootstrap C generation job
GENC77	CFT77 generation job
GENCAL	CAL and system texts generation job
GENCC	C generation job
GENCFT	CFT generation job
GENCOS	COS generation job
GENCOSL	COS libraries generation job
GENCOSU	COS utilities generation job
GENCSM	CSIM generation job

Decks	Routines
GENDBUG	Debuggers generation job
GENDIA	On-line diagnostics generation job
GENDSDK	Deadstart expander disk creation job
GENDSTP	Deadstart expander tape creation job
GENGOS	GOS utilities generation job
GENIOS	IOS generation job
GENLIB	Libraries generation job
GENMIG	COS pre-migration generation job
GENMUL1	Multitasking libraries (part 1) generation job
GENMUL2	Multitasking libraries (part 2) generation job
GENMULT	Multitasking libraries generation job
GENNCC	C 4.0 generation job
GENOCAL	OLDCAL and system tests generation job
GENPRD1	Products (part 1) generation job
GENPRD2	Products (part 2) generation job
GENPROC	PROCLIB generation job
GENPSC	Pascal and $PSCLIB generation job
GENRDM	RDM utilities generation job
GENSKOL	SKOL, SKOLREF and SKOLTXT generation job
GENSLM1	Superlink multitasking libraries (part 1) generation job
GENSLM2	Superlink multitasking libraries (part 2) generation job
GENSLMU	SUPERLINK multitasking SEGLDR and definition files generation job
GENSLS1	SUPERLINK stack libraries (part 1) generation job
GENSLS2	SUPERLINK stack libraries (part 2) generation job
GENSM45	COS Table Descriptions Internal Reference Manual generation job
GENSTC1	Stack libraries (part 1) generation job
GENSTC2	Stack libraries (part 2) generation job
GENSTCK	Stack $SYSDEF and $UTLDEF generation job
GENTAPE	Tape utilities generation job
GENC77	Temporary CFT77 generation job
GENTCAL	Temporary CAL and system texts generation job
GENTLIB	Temporary libraries generation job
GENTOOL	Tools generation job
GENTPSC	Temporary Pascal and $PSCLIB generation job
GENTUTL	Temporary utilities generation job
GENUTL	Utilities generation job
JSYSDIR	System directory job
PROCLIB	PROCLIB
PROCLIST	PROCLIB listing

B.14 HPMGRPL

All decks within HPMGRPL are built into $UTLIB. All routines in HPMGRPL are written in CAL.

B.15 IOLIBPL

All decks within IOLIBPL comprise $IOLIB. Deck EOF1 separates the routines written in CAL (first file) from those written in Fortran (second file).

B.16 IOPPL

See the I/O Subsystem (IOS) Administrator's Guide, publication SG-0307, for complete information on IOPPL.

B.17 LDRPL

Decks	Routines		
ADSTAPE	ADSTAPE		
BF	BUILD		
BIND	BIND		
L	LDR		
LDR2	LD2		
SEGLDR-->SEGLDR*	SEGLDR	SEGRLS	SEGRLS

B.18 MISCPL

All decks within MISCPL are built into $UTLIB. Deck EOF1 separates the routines written in CAL (first file) from those written in Fortran (second file).

B.19 MULTIPL

All decks within MULTIPL are built into $UTLIB. Deck EOF1 separates the routines written in CAL (first file) from those written in Fortran (second file).

B.20 PDSLBPL

All decks within PDSLBPL comprise $PDSLIB. An EOF in the ASCII deck separates the routines written in Fortran (first file) from those written in CAL (second file).

B.21 SCILBPL

All decks within SCILBPL comprise $SCILIB. Deck EOF1 separates the routines written in CAL (first file) from those written in Fortran (second file).

B.22 SIDPL

Decks	Routines
BEGINSID-->ENDSID	$SID
DBGHLPF, DBGHLPJ	$DBHELP

B.23 SKOLPL

Decks	Routines
SKOL	SKOL
SKOLREF	SKOLREF
SKOLTXT	SKOLTXT

B.24 SLLIBPL

Deck	Routine
CALSTART-->CALEND	$SLLIB
CFTSTART-->CFTEND	$SLLIB
CSTART-->CEND	$SLLIB
PASSTART-->PASEND	
FRLS	FRLS
FSELECT	FSELECT
HAACDEF	AACDEF
RELEASE	RELEASE
REWIND	REWIND
SLSUB	SLSUB

B.25 SYSDFPL

Decks	Routines
ST	$SYSTXT/$SYSDEF
TT	$TMPDEF

B.26 SYSLBPL

All decks within SYSLBPL comprise $SYSLIB. Deck EOF1 separates the routines written in CAL (first file) from those written in Fortran (second file).

B.27 TARGPL

All decks within TARGPL are built into $UTLIB. All routines in TARGPL are written in CAL.

B.28 TBMGRPL

All decks within TBMGRPL are built into $UTLIB. All routines in TBMGRPL are written in CAL.

B.29 TEDIPL

All decks within TEDIPL comprise TEDI. An EOF in the IZEXIT deck separates the routines written in Fortran (first file) from those written in CAL (second file).

B.30 TOOLPL

Decks	Routines
BLOCK	BLOCK
FTREF-->FTREF*	FTREF
MODSEQ	MODSEQ
MODSET	MODSET
NOTE	NOTE
PERFMON	PERFMON
SPAWN	SPAWN
STEP	STEP
SYSREF	SYSREF
UNBLOCK	UNBLOCK

B.30.1 SOFTWARE TOOLS (IN TOOLPL)

Routine names differing from the deck are shown in parentheses.

Decks	Decks
$PDDERR	PRTEV
ADMIN	QSTALIB - assembly version of boot library
AR	QSTAR - CFT version of AR
ASCII	QSTCAT - CFT version of CAT
CAT	QSTFFD - CFT version of FFIND
CH	QSTFLD - CFT version of FIELD
COMM	QSTFLIB - CFT version of boot library
CPRESS	QSTGET - CFT version of GET
CRT	QSTLPR - CFT version of LPR
CRYPT	QSTPR - CFT version of PR
DATE (DATIM)	QSTR41 - CFT version of RATP1
DC	QSTR42 - CFT version of RATP2
DELTA	QSTRM - CFT version of RM
DETAB	QSTSUI
DIFF	QSTYECH - CFT version of YECH (echo)
ECHO (YECH)	RATP1
ED	RATP2
EDITBA	RATLIB ($RATLIB, $RATDEF)

Decks	Decks
ENTAB	RC
EPITOME	REV
EXPAND	RM
FB	RTR
FFIND	SDF
FIELD	SEDIT
FIND	SETCOM
FORMAT	SETERR
GET	SETEV
GETEV	SHELL
INCLUD	SHEND
JCLSET	SHOW
KWIC	SORT (YSOR)
LAM	SPLIT
LEXLB,TOYSLIB ($LEXLB)	TABCAL
LL	TAIL
LPR	TDGROFF
LRGEN	TDGXTOC ($TDGTOC, $WTOOLS)
MACRO	TEE
MCOL	TR
MIV	TSORT
MV	UNIQ
OS	UNROT
PL	XREF
PPAUDPL	WC
PR	YACLR
YYPLB ($YYPLB)	

B.31 UPDPL

Decks	Routines
AUDPL-->AUDPL*	AUDPL
MODECKS-->MODECKS*	MODECKS
UPDATE-->UPDATE*	UPDATE
UPIC	UPIC

B.32 UTILPL

Decks	Routines
COMPARE	COMPARE
COPYD	COPYD
COPYF	COPYF

Decks	Routines
COPYNF	COPYNF
COPYR	COPYR
COPYU	COPYU
DEBUG	DEBUG
DSDUMP	DSDUMP
FLODUMP	FLODUMP
ITEMIZE	ITEMIZE
MTDUMP-->MTDUMP*	MTDUMP
PERFF,PERFS	$PERFLB
QUERY	QUERY
SKIPD	SKIPD
SKIPF	SKIPF
SKIPR	SKIPR
SKIPU	SKIPU
SORT	SORT
SPY	SPY
TARGET	TARGET
UNB	UNB

B.33 XMPPL

Decks	Routines
AHT	AHT
ARB	ARB
ARM	ARM
BRB	BRB
CMP	CMP
CMX	CMX
GTH	GTH
IBZ	IBZ
MIT	MIT
SFA	SFA
SFM	SFM
SFR	SFR
SIS	SIS
SR3	SR3
SRA	SRA
SRB	SRB
SRL	SRL
SRS	SRS
STAN	STAN
SVC	SVC
TRB	TRB
VPP	VPP
VRA	VRA
VRL	VRL

Decks	Routines
VRN	VRN
VRR	VRR
VRS	VRS
VRX	VRX
WRITEDS	WRITEDS

INDEX

READER'S COMMENT FORM

COS Operational Procedures Reference Manual SM-0043 G

Your reactions to this manual will help us provide you with better documentation. Please take a moment to check the spaces below, and use the blank space for additional comments.

1) Your experience with computers: ____ 0-1 year ____1-5 years ____5+ years
2) Your experience with Cray computer systems: ____0-1 year ____ 1-5 years ____5+ years
3) Your occupation: ____ computer programmer ____ non-computer professional
____ other (please specify): ______________________________
4) How you used this manual: ____ in a class ____as a tutorial or introduction ____ as a reference guide
____ for troubleshooting

Using a scale from 1 (poor) to 10 (excellent), please rate this manual on the following criteria:

5) Accuracy _____
6) Completeness ____
7) Organization ____
8) Physical qualities (binding, printing) ____
9) Readability ____
10) Amount and quality of examples ____

Please use the space below, and an additional sheet if necessary, for your other comments about this manual. If you have discovered any inaccuracies or omissions, please give us the page number on which the problem occurred. We promise a quick reply to your comments and questions.

Name ____________________________
Title ____________________________
Company ________________________
Telephone _______________________
Today's Date _____________________

Address _________________________
City ____________________________
State/ Country ____________________
Zip Code ________________________

CUT ALONG THIS LINE

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO 6184 ST PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY RESEARCH, INC.

Attention: PUBLICATIONS
1345 Northland Drive
Mendota Heights, MN 55120

FOLD

READER'S COMMENT FORM

COS Operational Procedures Reference Manual SM-0043 G

Your reactions to this manual will help us provide you with better documentation. Please take a moment to check the spaces below, and use the blank space for additional comments.

1) Your experience with computers: ____ 0-1 year ____1-5 years ____5+ years
2) Your experience with Cray computer systems: ____0-1 year ____ 1-5 years ____5+ years
3) Your occupation: ____ computer programmer ____ non-computer professional
____ other (please specify): ____________________________
4) How you used this manual: ____ in a class ____as a tutorial or introduction ____ as a reference guide
____ for troubleshooting

Using a scale from 1 (poor) to 10 (excellent), please rate this manual on the following criteria:

5) Accuracy _____
6) Completeness ____
7) Organization ____
8) Physical qualities (binding, printing) ____
9) Readability ____
10) Amount and quality of examples ____

Please use the space below, and an additional sheet if necessary, for your other comments about this manual. If you have discovered any inaccuracies or omissions, please give us the page number on which the problem occurred. We promise a quick reply to your comments and questions.

Name ______________________
Title ______________________
Company ______________________
Telephone ______________________
Today's Date ______________________

Address ______________________
City ______________________
State/ Country ______________________
Zip Code ______________________

CUT ALONG THIS LINE

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO 6184 ST PAUL, MN

POSTAGE WILL BE PAID BY ADDRESSEE

CRAY RESEARCH, INC.

Attention: PUBLICATIONS
1345 Northland Drive
Mendota Heights, MN 55120

FOLD