MANAGING DATASETS AND PROGRAMS
AT NCAR: The Mass Storage Subsystem

Authors: Dan Anderson
Mike Ernst
Astrik Deirmendjian
Editor: Gregory R. McArthur

SCIENTIFIC COMPUTING DIVISION
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH
BOULDER, COLORADO
This document describes how to manage your datasets and programs at NCAR using the Scientific Computing Division's Mass Storage Subsystem (MSS). It is a totally new user's guide designed to provide you with the information you need to make effective use of the CRAY-1A computers, the AMPEX Terabit Memory System, the IBM 4341 computing system, and the various physical storage devices (e.g. 1/2-inch tape, disks, etc.) which comprise the MSS. Liberal use of "how-to" examples and necessary Job Control Language (JCL) statements should help you become familiar with various techniques for managing your datasets.

ACKNOWLEDGEMENTS

The creation of this new user's guide has been a collaborative SCD effort. Marc Nelson of the Systems Section contributed documentation for the new SCD software which links the CRAY-1A disks to 1/2-inch tape drives and also verified the technical aspects of in-house software which is described in this guide. In addition, Marc conceived the cover illustration for this guide, lending much needed coherency to the diagram representing the assemblage of machines which make up the Mass Storage Subsystem. Mary Trembour of the Operations Section provided substantive input on the various ways to archive datasets and manage your programs on the AMPEX Terabit Memory System. Other individuals contributed their time and expertise to this guide by providing software documentation and examples to make your use of this guide easier. Chief among these contributors are David Kennison, Dennis Joseph, Stu Henderson, Joy Choy, David Kitts, and Bonnie Gacnik. Finally, the meticulous and insightful proofreading of the entire document by Barb Horner, Consulting Project Leader, is appreciatively acknowledged by all who worked on this user's guide.
# Table of Contents

**MANAGING DATASETS AND PROGRAMS AT NCAR: The Mass Storage Subsystem**

## CHAPTER 1: AN OVERVIEW OF THE MASS STORAGE SUBSYSTEM

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1-1</td>
</tr>
<tr>
<td>Components of the Mass Storage Subsystem at NCAR.</td>
<td>1-2</td>
</tr>
<tr>
<td>The AMPEX Terabit Memory System</td>
<td>1-3</td>
</tr>
<tr>
<td>Twin CRAY-1A Computers</td>
<td>1-4</td>
</tr>
<tr>
<td>CRAY-1A 1/2-inch Tapes</td>
<td>1-4</td>
</tr>
<tr>
<td>IBM 4341 1/2-inch Tapes</td>
<td>1-4</td>
</tr>
<tr>
<td>The NCAR Local Network</td>
<td>1-5</td>
</tr>
</tbody>
</table>

## CHAPTER 2: TECHNICAL DESCRIPTIONS AND DEFINITIONS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>2-1</td>
</tr>
<tr>
<td>The AMPEX Terabit Memory System (TBM)</td>
<td>2-1</td>
</tr>
<tr>
<td>Dataset Definitions</td>
<td>2-2</td>
</tr>
<tr>
<td>Labeled Datasets</td>
<td>2-2</td>
</tr>
<tr>
<td>Non-NCAR Labeled Datasets</td>
<td>2-3</td>
</tr>
<tr>
<td>Unlabeled Datasets</td>
<td>2-3</td>
</tr>
<tr>
<td>Types of Datasets</td>
<td>2-3</td>
</tr>
<tr>
<td>CRAY-1A Datasets</td>
<td>2-3</td>
</tr>
<tr>
<td>CRAY-1A Dataset Names</td>
<td>2-3</td>
</tr>
<tr>
<td>PERMANENT Datasets</td>
<td>2-4</td>
</tr>
<tr>
<td>LOCAL Datasets</td>
<td>2-4</td>
</tr>
<tr>
<td>TBM Datasets</td>
<td>2-4</td>
</tr>
<tr>
<td>The TBM Envelope</td>
<td>2-4</td>
</tr>
<tr>
<td>TBM Dataset Conventions</td>
<td>2-4</td>
</tr>
<tr>
<td>Special Problems with TBM Dataset and File Names</td>
<td>2-5</td>
</tr>
<tr>
<td>Dataset Organization on the TBM</td>
<td>2-5</td>
</tr>
<tr>
<td>TBM Physical Record Structure</td>
<td>2-5</td>
</tr>
<tr>
<td>Blocked Record Format</td>
<td>2-5</td>
</tr>
<tr>
<td>Error Detection on the TBM</td>
<td>2-6</td>
</tr>
<tr>
<td>IBM Datasets</td>
<td>2-6</td>
</tr>
<tr>
<td>Dataset Names</td>
<td>2-6</td>
</tr>
<tr>
<td>IBM File Names</td>
<td>2-6</td>
</tr>
<tr>
<td>Other Dataset Formats</td>
<td>2-7</td>
</tr>
<tr>
<td>Segmented Dataset Formats</td>
<td>2-7</td>
</tr>
<tr>
<td>File and Record Access on MSS Devices</td>
<td>2-8</td>
</tr>
<tr>
<td>CRAY-1A Disk</td>
<td>2-8</td>
</tr>
<tr>
<td>1/2-inch Tape</td>
<td>2-8</td>
</tr>
<tr>
<td>ACQUIRE and DISPOSE JCL Statements</td>
<td>2-8</td>
</tr>
</tbody>
</table>
CHAPTER 3: MOVING DATASETS

Introduction.............................................................. 3-1
Staging Jobs on the CRAY-1A Computers.................................. 3-1
CRAY-1A JCL Used to Move Datasets.................................... 3-2
  DISPOSE................................................................ 3-2
  ACQUIRE................................................................ 3-2
  Mainframe Identifier.................................................. 3-2
  Dataset Format......................................................... 3-3
  Disposition Code....................................................... 3-3
  Summary Tables........................................................ 3-3
  Additional ACQUIRE and DISPOSE Parameters...................... 3-6
Storing and Retrieving Datasets on the TBM............................ 3-6
  CRAY-1A Datasets.................................................... 3-6
  CDC 7600 Datasets.................................................. 3-7
Staging Tapes............................................................ 3-8
  Cray Direct Tape Connection....................................... 3-8
  Terms Used............................................................ 3-8
  A Note on Generic Resources...................................... 3-9
  TEXT Field Keywords............................................... 3-10
  COS-Supported Keywords.......................................... 3-14
  Examples Using Cray Direct Tape.................................. 3-16
Moving Datasets to NCAR Local Network Nodes........................ 3-17
  Storing and Retrieving IBM-CMS Files.............................. 3-17
  CMS Text and Program Files........................................ 3-17
  Binary CMS Files on the TBM...................................... 3-19
Disposing Datasets to Your RJE Site via RSCS.......................... 3-21
Disposing Datasets to the MDCOMP.................................... 3-21
Moving Datasets to the DICOMED....................................... 3-21
  Plot or Metacode Files from the TBM............................. 3-22
  Print Character Data from Tape on Fiche......................... 3-22
Moving Datasets to Other Nodes on the NCAR Local Network......... 3-22

CHAPTER 4: INPUT/OUTPUT AND DATASET CONVERSION ROUTINES

Introduction.............................................................. 4-1
Routines Described...................................................... 4-1
Section I: I/O Routines................................................ 4-2
  BRANIO: Direct (Random) I/O..................................... 4-3
  RDTAPE, WRITAPE, IOWAIT........................................ 4-5
  Subroutine UBLOCK................................................ 4-9
  Subroutine UZBLOCK............................................... 4-11
  RPTIN, RPTOUT..................................................... 4-14
Section II: Data Conversion Routines.................................. 4-19
  Routines for Converting CDC 7600 TEM Volumes.................. 4-19
    TEMCONV......................................................... 4-19
    REFRMT........................................................... 4-23
  Routines for the Conversion of CDC 7600 Data.................... 4-25
    SCONV............................................................. 4-25
    DSEG.............................................................. 4-29
Useful Tools for General Conversion Problems........................ 4-30
  GBYTES-GBYTE & SBYTES-SBYTE.................................. 4-30

Table of Contents -2- October 1983
### CHAPTER 5: EFFECTIVE USE OF THE TBM

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>5-1</td>
</tr>
<tr>
<td>Background Information</td>
<td>5-1</td>
</tr>
<tr>
<td>Reading and Writing Your VSNs.</td>
<td>5-1</td>
</tr>
<tr>
<td>What is Involved in a TBM Tape Mount?</td>
<td>5-2</td>
</tr>
<tr>
<td>Dedicated Reels</td>
<td>5-2</td>
</tr>
<tr>
<td>Large VSNs</td>
<td>5-2</td>
</tr>
<tr>
<td>Archiving Datasets on the TBM</td>
<td>5-3</td>
</tr>
<tr>
<td>General Considerations</td>
<td>5-3</td>
</tr>
<tr>
<td>VSN Purge Policy</td>
<td>5-3</td>
</tr>
<tr>
<td>Exemptions</td>
<td>5-4</td>
</tr>
<tr>
<td>Definitions for Dedicated Reel Users</td>
<td>5-4</td>
</tr>
<tr>
<td>Log Tape</td>
<td>5-4</td>
</tr>
<tr>
<td>Production (prod) Tape</td>
<td>5-4</td>
</tr>
<tr>
<td>Volume Serial Name (VSN)</td>
<td>5-4</td>
</tr>
<tr>
<td>Dedicated Reel (MVN)</td>
<td>5-4</td>
</tr>
<tr>
<td>Joint User Dedicated Reel</td>
<td>5-5</td>
</tr>
<tr>
<td>Archival Dedicated Reel</td>
<td>5-5</td>
</tr>
<tr>
<td>On-line System</td>
<td>5-5</td>
</tr>
<tr>
<td>Indirect Descend</td>
<td>5-5</td>
</tr>
<tr>
<td>VSN Moves</td>
<td>5-5</td>
</tr>
<tr>
<td>Guidelines for Dedicated Reel Use</td>
<td>5-5</td>
</tr>
<tr>
<td>Utilities for TBM Dataset Maintenance</td>
<td>5-6</td>
</tr>
<tr>
<td>UPKEEP</td>
<td>5-7</td>
</tr>
<tr>
<td>TBMVSN</td>
<td>5-8</td>
</tr>
<tr>
<td>DTILIST: On-line Dedicated Reel Information</td>
<td>5-9</td>
</tr>
<tr>
<td>Some DOs and DON'Ts for the TBM</td>
<td>5-9</td>
</tr>
<tr>
<td>Further Assistance</td>
<td>5-11</td>
</tr>
</tbody>
</table>

### CHAPTER 6: JOB DECK SETUP EXAMPLES

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>6-1</td>
</tr>
<tr>
<td>Command Syntax Review</td>
<td>6-1</td>
</tr>
<tr>
<td>Section I: Moving Datasets between the CRAY-1A Disks and the TBM.</td>
<td>6-3</td>
</tr>
<tr>
<td>ACQUIRE a CRAY-1A Dataset from the TBM</td>
<td>6-3</td>
</tr>
<tr>
<td>DISPOSE a CRAY-1A Dataset to the TBM</td>
<td>6-3</td>
</tr>
<tr>
<td>ACQUIRE TBM Dataset Written on the CDC 7600</td>
<td>6-4</td>
</tr>
<tr>
<td>Section II: Moving Datasets between the CRAY-1A Disks and the IBM 4341</td>
<td>6-5</td>
</tr>
<tr>
<td>DISPOSE a CRAY-1A Dataset to the IBM 4341</td>
<td>6-5</td>
</tr>
<tr>
<td>ACQUIRE a Dataset from the IBM 4341</td>
<td>6-5</td>
</tr>
<tr>
<td>Section III: Moving Datasets from CRAY-1A Tape to CRAY-1A Disks</td>
<td>6-6</td>
</tr>
<tr>
<td>Reading a CRAY-1A Tape to a CRAY-1A Disk</td>
<td>6-6</td>
</tr>
<tr>
<td>Variations on the Theme</td>
<td>6-6</td>
</tr>
<tr>
<td>Section IV: Moving Datasets from CRAY-1A Disks to CRAY-1A Tape</td>
<td>6-8</td>
</tr>
<tr>
<td>Section V: Setting up Composite Jobs to Stage Datasets</td>
<td>6-10</td>
</tr>
</tbody>
</table>
INTRODUCTION

This guide provides you with information and assistance in managing data and programs using the Mass Storage Subsystem (MSS) at NCAR.

The Mass Storage Subsystem is a collection of software and hardware available at NCAR designed specifically for data management. In this context, data (and another term, datasets) can refer to program files which may be either source or object codes, or data files which are usually in binary form. Our primary concern is to describe how you may access and store datasets using the components of the Mass Storage Subsystem. In particular, this guide discusses using the Terabit Memory System (TBM) and the tape drives linked directly to the CRAY,CA computer.

Frequent use of examples and the associated command syntax are provided to aid you in learning how to perform the most common tasks associated with moving, storing, and retrieving programs and data within the Scientific Computing Division (SCD). Additional helpful information covering more technical aspects of the MSS and its use may be found in the appendices to this guide. A glossary of terms used in this guide is also included for your reference.

It is assumed that you have read and are reasonably familiar with the information contained in the following three documents:

- Batch Computing on the CRAY-1 (TN-179+IA)
- NCAR Local Network (TN-181+IA) and,
- The IBM 4341: Gateway to NCAR Computers (TN-196+IA)

These documents are all available from the Scientific Computing Division (SCD) Documentation Project.
Large amounts of data are brought to SCD from many different institutions and research organizations pursuing investigations into the atmospheric sciences. Even greater amounts are generated by our users during the course of their computations. No matter what kind of computing you do at NCAR, you will be making use of the Mass Storage Subsystem to manage your datasets and programs. The major components of the Mass Storage Subsystem currently include the following:

1. An AMPEX Terabit Memory System (TBM) mass storage device
2. Twin CRAY-lA computers
3. Two IBM 4341 computers, and
4. The NCAR Local Network

The two CRAY-lA computers have fixed links to the TBM and, in fact, provide the only direct communication with the TBM. The CRAY-lA machines also have a direct link to tape drives via locally-developed NCAR software which is not supported by Cray Research, Inc. (CRI).

An additional route which your data may take is to move through the NCAR Local Network. Using the NCAR Local Network, you may access any computer (referred to as a 'node') attached to the NCAR Local Network. The NCAR Local Network is fully documented in the user's guide entitled the NCAR Local Network (TN-181+IA) which is available from the SCD Documentation Project.
Figure 1.1 provides an overview of the Mass Storage Subsystem at NCAR.

The AMPEX Terabit Memory System was installed in March of 1976. Designed to provide state-of-the-art storage technology for our users, it has rapidly become the most widely used component of the subsystem. A complete description of the current configuration of the AMPEX Terabit Memory System may be found in Chapter 2.
NCAR's first CRAY-1A computing system (serial #3) was installed in July of 1977. The second CRAY-1A (serial #14) was installed in January 1983. For convenience of machine identification, the serial #3 machine is designated as 'CRAY,C1', and the serial #14 machine is known as 'CRAY,CA'. These naming conventions will be used throughout this guide.

The twin CRAY-1A computers are central to all data movement among these parts of the subsystem as well as to all other computing machines on the NCAR Local Network. The CRAY,C1 and CRAY,CA each have a memory of 1,048,576 64-bit words. Associated with each CRAY-1A is a bank of 16 disk drives which provide 2.4 billion bits of rotating storage each. Direct communication with the TBM is handled through the CRAY-1A computers, while the tape drives attached to the twin CRAY-1A mainframes or to the IBM 4341 may be accessed in various ways by using the appropriate software tools mentioned below and described in Chapter 3.

At this printing, only one of the CRAY-1A computers is attached to 1/2-inch tape drives. The CRAY,CA computer has access to high speed, high density 1/2-inch tapes. The CRAY,C1 computer will soon have similar tape access capabilities. You may access datasets on these tapes by using NCAR-developed software routines which interface with the Cray control statements ACQUIRE and DISPOSE. Both CRAY-1A computers will be able to access tapes of virtually all recording formats (NRZ and BIPHASE), densities (200 to 6250 bpi), and number of tracks (7† or 9). These tapes provide immediate entry to the MSS for those of you who bring data in various formats to NCAR for processing. Details on how to move data from your 1/2-inch tapes onto the TBM or to the CRAY-1A disks to be accessed by the CRAY-1A computers using ACQUIRE and DISPOSE will be discussed in detail in Chapter 3.

Tape drives attached to the IBM 4341 minicomputers may be accessed by the CRAY-1A computers using software developed by Cray Research, Inc. (CRI). The Data Management Support Processor (DMSP) software is designed to permit you to access datasets on the IBM 4341 tape drives and send them to/from the TBM via the CRAY-1A computers. DMSP is discussed in detail in Appendix C of this guide. This software is primarily restricted for use by the Systems Group of SCD and for this reason will not be discussed in detail here.

Note: If you have data on a 7-track tape, you may copy it to a 9-track tape on the IBM 4341 machines.

† Deferred implementation on the CRAY,CA direct tape link.
The NCAR Local Network is composed of a number of computer systems loosely coupled using the Network Systems Corporation's (NSC) network adapter boxes. "Loose coupling" means that each computer system operates independently of the others. Each adapter is linked to a coaxial cable or trunk which permits it to communicate with other adapters on the same trunk. Thus, any two computer systems can potentially communicate with each other if their adapters are on the same trunk. Detailed information on the use of the NCAR Local Network may be found in the user's guide entitled the NCAR Local Network (TN-181+IA) available from the Documentation Project.
CHAPTER 2: TECHNICAL DESCRIPTIONS AND DEFINITIONS

INTRODUCTION

This chapter provides you with a detailed description of the Terabit Memory System (TBM), its configuration, and describes datasets as they are defined for the different nodes of the Mass Storage Subsystem (MSS). The details of access, format, labeling, and naming requirements and conventions for datasets on the various components of the MSS are also covered in this chapter.

THE AMPEX TERABIT MEMORY SYSTEM (TBM)

Since the TBM is one of the most intensively used components of the Mass Storage Subsystem, it merits some detailed discussion. Most of you will make use of the TBM in reading and/or writing your datasets onto SCD computers.

Your datasets are stored on the TBM by means of 2-inch video tape rather than the standard 1/2-inch magnetic tape. Information is not recorded as individual magnetic spots or bits as on 1/2-inch tape. Rather, the entire bit stream coming from the CRAY-1A machines modulates an FM carrier signal. Since the TBM uses video tape instead of 1/2-inch magnetic tape, the resulting FM-modulated carrier signal is written on the TBM tape in the same way that video TV signals are recorded. This has the net effect of reducing the bit-stream into a frequency spectrum which is much more reliable in reducing the potential for loss of bit(s) and the attendant information. The error rate is one error in ten billion bits.
The current configuration of the AMPEX Terabit Memory System is given in Table 2.1 below.

<table>
<thead>
<tr>
<th>Table 2.1</th>
<th>AMPEX Terabit Memory System Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Storage Section</strong></td>
<td></td>
</tr>
<tr>
<td>16 Terabit Memory Tape Drives</td>
<td>43 billion bits per reel</td>
</tr>
<tr>
<td></td>
<td>Data transfer rate: 700 kilobytes/s</td>
</tr>
<tr>
<td></td>
<td>Average access time: 17 sec.</td>
</tr>
<tr>
<td>5 Transport Drivers</td>
<td>Direct all transport and internal data activity.</td>
</tr>
<tr>
<td>3 Data Channels</td>
<td>Perform simultaneously independent read/write operations at 700 kilobytes/s.</td>
</tr>
</tbody>
</table>

| **Communication and Control Section** | |
| 1 Storage Control Processor (PDP 11/35) | Data transfer requests initiated by host are interpreted by the SCP which maintains an internal master file directory of all data files, on-line and off-line. |
| 1 Master File Directory | Capable of supporting 100,000 datasets. |
| 2 Channel Interface Units (PDP 11/05) | Executes data transfer between host and data channels. |

**DATASET DEFINITIONS**

**Labeled Datasets** Most of your existing TBM datasets will be labeled according to standards proposed by the American National Standards Institute-Conference on Data Systems Languages (ANSI-CODASYL). Prior to April 7, 1983, TBM datasets generated on NCAR’s CDC 7600 were also given ANSI-CODASYL labels. TBMCONV, which is described in Chapter 4 of this guide, is used to strip this information from these datasets. An ANSI-CODASYL labeled dataset is a dataset containing labeled files. Labeled files are groups of three files; the first file contains the file header label information and the volume header information, the second contains...
the data, and the third file contains the trailer label information.

TBM/CRAY Interface software (residing on the CRAY-1A mainframes) uses an SCD-specific set of software tools which permit ANSI-CODASYL labeled datasets to be read from the TBM. The operation is entirely transparent to the user.

Non-NCAR Labeled Datasets

If you bring a dataset to NCAR from another installation you can enter that dataset into the system and move it to the machine of your choice. **NOTE:** you may have to go in with your own program to decode a labeled file structure. If you have a choice you should bring unlabeled tapes to NCAR.

Unlabeled Datasets

Since dataset movement onto or off of the TBM is handled through the CRAY-1A computers, all labeled datasets must be stripped of their labeling information. The Cray operating system does not recognize labeled datasets. In fact, SCD cannot read 1/2-inch labeled tape on the CRAY-1A machines without special processing.

TYPES OF DATASETS

CRAY-1A Datasets

CRAY-1A datasets are the largest units of data handled by the CRAY-1A computers during execution of your computing job. They are also the units transferred by the CRAY-1A mainframes to the TBM, the IBM 4341, or other computing machines within SCD.

At this time the defacto definition of a dataset is the CRAY-1A dataset. We will discuss all other datasets in terms of their mapping into CRAY-1A datasets. CRAY-1A datasets are not labeled, that is, there is no dataset header label record nor are the files within a dataset labeled.

**NOTE:** Any dataset generated on the old NCAR CDC 7600 using CONV=TB or STAGEOUT=MDT will be a labeled dataset. The software to convert these datasets to unlabeled CRAY-1A datasets is called automatically when you specify a dataset format of DF=76 as one of the parameters to the ACQUIRE command. Tape datasets written without CONV=TB were NOT labeled on the CDC 7600. For more information on CDC 7600 generated tape and TBM datasets, please refer to the examples in Chapter 6 of this guide.

CRAY-1A Dataset Names

A permanent dataset on either CRAY-1A computer may have a 1-to-15 character name. The first character of that name may be an alphabetic character or one of the following characters: "a", "s", or "8". The characters immediately following the first character of a name may be alphabetic...
or numeric. Use of dataset names of 6 or fewer characters is required, as the TBM will reject any dataset with a name longer than 6 characters.

**PERMANENT Datasets**

Permanent datasets are created on the CRAY-1A disks and remain "on-line" for a period of hours, depending upon the disk load. Permanent datasets will be "aged off" or removed from the CRAY-1A disks periodically based upon the disk loading.

**LOCAL Datasets**

Local datasets are those which are created or logically attached by a job running on the CRAY-1A computers. They exist only for the duration of job execution.

**GENERAL NOTE:** Due to the unusually heavy loading of the disks on the CRAY-1A systems, no permanent storage of datasets is permitted on the CRAY-1A mainframes. Any datasets you wish to save should be sent to the TBM.

**TBM DATASETS**

**The TBM Envelope**

Any dataset placed on the TBM is kept in "native" format. Native refers to the format of the file as it exists when it is moved onto the TBM. At the moment, that native format is CRAY-1A format, since the CRAY-1A computers are the only machines which access the TBM. As your dataset is being written onto the TBM, the TBM automatically places a labeled header- and end-block of information around precise 122,880 byte segments of your dataset. A labeled header segment is placed in front of the first 122,880 byte segment and a trailer-label segment is placed after the last data segment. The header-label segment contains information about your dataset (such as dataset name, password(s), etc.). The trailer-label segment contains checksums of the data when it was written, ensuring that any errors contained in the data will be detected upon retrieval.

**NOTE:** The creation of the TBM envelope is an entirely transparent process. It does not affect the integrity of your dataset and "disappears" when you take your dataset off of the TBM and place it back on one of the CRAY-1A computers.

**TBM Dataset Conventions**

A TBM dataset (also known as a VSN or Volume Serial Name) is the unit moved to and from the TBM and may contain one or more files. The name contains six or fewer characters and identifies the volume to be read or written. Naming conventions are the same as for datasets on the CRAY-1A, except that the first character may be numeric. Starting VSNs with the same two or three letters followed by a number series helps both you and SCD Operations keep track of them. The use of write passwords prevents
overwriting by others.

Special Problems with IBM Dataset and File Names

Some of you may have datasets which were written on the CDC 7600 computer. The CDC 7600 mainframe (now decommissioned) created named files within datasets. You may still access files from these datasets by their name using NCAR-developed JCL on the CRAY-1A computers. If the 17 character (maximum) name contains special characters (e.g. a period ( . )), the name referenced in your JCL must be placed within single quote marks ( ' ). Chapter 4 of this guide discusses most of the software where this is applicable.

Dataset Organization on the TiM

The organization of datasets to be moved onto or off of the TiM is critical to the effective working of the TiM. Recall that a dataset is a collection of physical records, normally a blocked representation of a collection of logical records. The collection of records in a dataset may be partitioned into files; files are delineated with filemarks which may be searched or tested for by appropriate I/O commands.

TEM Physical Record Structure

The software supported physical record structure of the TiM is a blocked record structure. The TiM blocks are fixed length, 122,880 byte records.

NOTE: All datasets written on obsolete machines will be supported. For example, TLIB volumes written on the CDC 7600 can still be read and translated on the CRAY-1A machines.

Blocked Record Format

BLOCKING refers to taking your record (referred to as the logical record) and dividing it into many smaller segments. Figure 2.1 illustrates how a large logical record is mapped into many physical record blocks.
The data integrity of the TBM is on the order of one error in 10 billion bits. This integrity is maintained by checksum information created and added into the bit stream by the computers driving the TBM. The TBM places this information in the trailer-label segment of the TBM envelope. Hence the checksum information becomes an integral part of the TEM dataset.

IBM DATASETS

Dataset Names Datasets named on the IBM 4341 computers consist of a filename and a filetype (see below), each of which is up to 8 alphanumeric characters long.

IBM File Names File names on the IBM 4341 are limited to 8 alphanumeric characters. In IBM terminology, this is known as the FILENAME (or FINM). The FILETYPE (or FLTY) may also consist of up to 8 alpha-numeric characters. For example, a dataset on the IBM 4341 would have the following name:

MYPROG2 INPUT2

where the filename is ‘MYPROG2' and the filetype is ‘INPUT2'. You may structure your dataset names on the IBM 4341 in such a way that the filename-filetype arrangement indicates what your dataset contains. For more information on naming datasets on the IBM 4341, please refer to the user's guide The IBM 4341: Gateway to NCAR Computers (TN-196+IA) which is available from the Documentation Project.
Other Dataset Formats

Segmented Dataset Formats

The most common format for 1/2-inch tapes is a segmented format. Segmentation pointers are blocking pointers created when a logical record consists of more than one physical record. There have been a few CDC 7600 TEM datasets written with embedded segmentation information (i.e. 1/2-inch tape images). If a dataset with this type of segmentation information is acquired on the CRAY-1A, the segmentation information should be removed with Cray library software routines (such as RPMRT for segmented TEM datasets and DSEG for segmented 1/2-inch tapes), since that form of tape organization is not useful on the CRAY-1A computers at NCAR. The only 1/2-inch tape writes that create segmented records are FORTRAN binary writes. See Chapter 4 for more detailed information on these tape format software routines.

Figure 2.2 shows a typical logical record-to-tape segmentation.

Figure 2.2
LOGICAL RECORD-TO-TAPE SEGMENTATION

LOGICAL RECORD

FIRST BLOCK

LAST BLOCK

FIXED TAPE BLOCKS
FILE AND RECORD
ACCESS ON MSS
DEVICES

CRAY-1A Disk As a general rule, direct access of a record or file on a
CRAY-1A disk should be performed on unblocked CRAY-1A da-
tasets, whereas sequential access may be efficiently per-
formed on either blocked or unblocked datasets.

1/2-inch Tape Datasets are typically written in sequential order on
1/2-inch tape. It is also possible to read files and
records sequentially "out of order" on 1/2-inch tape. It
is not possible to write 1/2-inch tape in other than a
sequential manner.

ACQUIRE and
DISPOSE JCL
Statements "ACQUIRE" and "DISPOSE" are CRAY-1A Job Control Language
(JCL) statements used to move your dataset onto or off of
the CRAY-1A computers. Each has a specific syntax which
permits you some degree of flexibility in moving your
data around. The complete list of parameters and key-
words associated with the ACQUIRE and DISPOSE statements
are found in Chapter 3 of this guide. Examples of how
to use this JCL are contained in Chapter 6.
INTRODUCTION

This chapter discusses dataset movement between the various elements of the Mass Storage Subsystem (MSS) and the commands used to accomplish these movements.

Recall that the CRAY-1A computers are central to all dataset movement among the components of the Mass Storage Subsystem. Both the IBM and high-speed 1/2-inch tape drives are connected to the CRAY-1A computers. IBM 4341 tape drives are available to the CRAY-1A machines via the Data Management Support Processor (DMSP) software developed by Cray Research, Inc., or the NCAR-developed Cray Direct Tape Connection software. The Cray Direct Tape Connection is discussed in this chapter. Information on DMSP may be found in Appendix C.

STAGING JOBS ON THE CRAY-1A COMPUTERS

Staging is a process of loading or unloading datasets to/from a disk file system from/to a remote computer file system.

As an example, suppose you have a program that requires a particular dataset. That dataset resides on machine "A" and your program resides on machine "B". Your program on machine "B" submits a job to one of the CRAY-1A computers which gets (or ACQUIRES) the dataset from machine "A" and sends (or DISPOSES) it to machine "B". The dataset is then resident on machine "B" and the program that reads it can then be executed. This process is called staging. On all machines except the IBM 4341, the dataset is staged directly to the machine's disks. Datasets moving to and from the IBM 4341 go through the "spool" file system.

Dataset staging is usually accomplished by a job run on one of the CRAY-1A computers (CA or CI), either by using the appropriate Job Control Language (JCL) statements before running your FORTRAN program or by calls to DISPOSE or ACQUIRE from within your FORTRAN program. The advantage of the latter procedure is that no data is staged unless your program actually begins to execute on one of the CRAY-1A computers (i.e. if your job doesn't compile correctly, then the data isn't loaded onto the CRAY-1A disk). The disadvantage is that this process may be more costly in terms of computer resources.
There are two commands, DISPOSE and ACQUIRE, which stage datasets to and from the CRAY-1A disk subsystem.

**DISPOSE**
This command provides a means of having a CRAY-1A dataset sent to a remote computer or storage subsystem. The form of the DISPOSE statement is:

```
DISPOSE,DN=dn,SDN=sdn,DC=dc,DP=df,MF=mf,ID=uid,TID=tid,
R=rd,W=wt,TEXT='text',WAIT,NOWAIT,DEFER,NRLS.
```

**ACQUIRE**
This command provides a means of getting a dataset from a remote computer/storage-system, making it a permanent dataset on the CRAY-1A disk and having it attached to your CRAY-1A job. The form of the ACQUIRE statement is:

```
ACQUIRE,DN=dn,PDN=sdn,DF=df,MF=mf,ID=uid,TID=tid,UQ,
R=rd,W=wt,M=mn,TEXT='text'.
```

**Mainframe Identifier**
The source or destination mainframe identifier, "MF=xx", may be specified in the following ways:

- TB-- The TBM
- MT-- Direct Tape Connected to CRAY, CA (serial #14)
- I2-- 1/2" tape mounts connected through IBM 4341 (DMSP)
- DI-- Dicomex plotting device-35mm, fiche and 16mm movie
- IO-- IBM 4341 (#1 machine)
- IA-- IBM 4341 (#2 machine)
- MC-- Modcom RJE station
- CA-- CRAY-1A serial #14
- C1-- CRAY-1A serial #3
- HA-- High Altitude Observatory PDP 11/70
- HV-- High Altitude Observatory VAX
- SB-- SCD System B PDP 11/70
- SA-- SCD System A PDP 11/70

**REMINDER:** You are reminded that the use of many of these mainframes requires the declaration of the "generic resource" on the JOB card. For example, use of the direct tape link on CRAY, CA requires the specification of *MT on the JOB card.
Dataset Format
Dataset format "DF=xx". DF= may have the following assignments:

TR -- Dataset is assumed to be in proper CRAY-1A format, no action is taken by the remote mainframe.
BB -- Dataset is put in CRAY-1A blocked format by the remote mainframe.
BD -- Dataset is transmitted as is by the remote mainframe.
CD -- Dataset is converted to ASCII and transmitted as is by the remote mainframe.
CB -- Dataset is converted to ASCII and then put in CRAY-1A blocked format by remote mainframe before transmission to one of the CRAY-1A computers.

Disposition Code
Disposition code "DC=xx". DC may have the following assignments:

ST-- Dispose on permanent file system of remote mainframe
PR-- Print on remote mainframe
PT-- Plot on remote mainframe
SP-- Dispose in spool file of remote machine—IBM only
MT-- Dispose or acquire from mag-tape on remote mainframe

Summary Tables
Table 3.1 shows the allowable parameter specifications for the mainframe identifier (MF=) versus the dataset format (DF=) parameters. A (+) indicates that the mainframe specified will accept the given DF= parameter. A (-) indicates that the mainframe specified will not accept the given DF= parameter.
Table 3.1
ACQUIRE and DISPOSE Conditions for MF= and DF= Parameters

The Mainframe identifier (MF=) is:
TB=Terabit Memory System (TBM)
MT=Cray Direct Tape Connection To CRAY, CA
I2=IBM 1/2-inch Tape Mounts via DMSP
IO=IBM 4341 (#1 machine)
IA=IBM 4341 (#2 machine)
Cl=CRAY-1A, Serial #3
CA=CRAY-1A, Serial #14
Dl=DICOMED Graphics Processors
MC=MODECOMP RJE Station

The Dataset Format (DF=) is:
TR=Dataset is assumed to be in proper CRAY-1A format. No action taken by remote computer.
BB=Dataset is transmitted in CRAY-1A blocked format by remote computer.
BD=Dataset transmitted 'as is' by remote computer.
CD=Dataset is converted to ASCII and transmitted 'as is' by the remote computer.
CB=Dataset is converted to ASCII and then put into CRAY-1A blocked format by remote mainframe.

<table>
<thead>
<tr>
<th>MF=</th>
<th>TB</th>
<th>MT</th>
<th>I2</th>
<th>IO</th>
<th>IA</th>
<th>Cl</th>
<th>CA</th>
<th>Dl</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF=</td>
<td>TR</td>
<td>(+) (+) (+) (-) (-) (+) (+) (-) (+)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB</td>
<td>(+) (+) (+) (+) (+) (+) (+) (+) (+)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BD</td>
<td>(+) (-) (-) (-) (-) (+) (+) (-) (+)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CD</td>
<td>(+) (-) (-) (-) (-) (+) (+) (-) (+)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CB</td>
<td>(+) (+) (+) (+) (+) (+) (+) (+) (+)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1= Regardless of what Dataset Format (DF=) parameter you specify for the TBM, it will assume a dataset format of (DF=TR).
2= All DF= parameters for the MC resource are assumed to be DF=CB. All other parameters are ignored.

Moving Datasets 3-4 October 1983
Table 3.2 shows the allowable parameter specifications for the mainframe identifier (MF=) versus the disposition code (DC=) parameters. The same (+) and (-) conventions hold for Table 3.2.

**Table 3.2**

ACQUIRE and DISPOSE Conditions for MF= and DC= Parameters

<table>
<thead>
<tr>
<th>MF= / DC=</th>
<th>TB</th>
<th>MT</th>
<th>I2</th>
<th>IO</th>
<th>IA</th>
<th>Cl</th>
<th>CA</th>
<th>D1</th>
<th>MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
</tr>
<tr>
<td>PR</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(-)</td>
<td>(+)</td>
<td>(+)</td>
</tr>
<tr>
<td>PT</td>
<td>(+)</td>
<td>(+)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(+)</td>
<td>(+)</td>
</tr>
<tr>
<td>SP</td>
<td>(+)</td>
<td>(+)</td>
<td>(-)</td>
<td>(+)</td>
<td>(+)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(+)</td>
</tr>
<tr>
<td>MT</td>
<td>(+)</td>
<td>(+)</td>
<td>(+)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(-)</td>
<td>(+)</td>
</tr>
<tr>
<td>BO</td>
<td>(+)</td>
<td>(+)</td>
<td>(-)</td>
<td>(+)</td>
<td>(+)</td>
<td>(-)</td>
<td>(-)</td>
<td>(+)</td>
<td>(+)</td>
</tr>
</tbody>
</table>

2 = Treated the same as PR.
The Mainframe identifier (MF=) is:

TB = Terabit Memory System (TBM)
MT = Cray Direct Tape Connection To CRAY,CA
T2 = IBM 1/2-inch Tape Mounts via DMSP
IO = IBM 4341 (#1 machine)
IA = IBM 4341 (#2 machine)
C1 = CRAY-1A, Serial #3
CA = CRAY-1A, Serial #14
D1 = MODCOMP Graphics Processors
MC = MODCOMP RJE Station

The Disposition Code (DC=) is:

ST = Put on permanent file system of remote mainframe
PR = Print on remote mainframe
PT = Plot on remote mainframe
SP = Plot on spool file of remote IBM 4341 only
MT = Put or get from mag-tape on remote mainframe
BO = Batch Output

Additional ACQUIRE and DISPOSE Parameters

"R=xx" -- 8 character read password—if written with dataset, you must use on ACQUIRE in order to obtain read permission.

"W=xx" -- 8 character write password—if already written on dataset, you must use it on the ACQUIRE in conjunction with "UO" in order to write on a dataset.

"ID=xxxxxxxxx" -- eight character ID assigned to CRAY-1A disk dataset to uniquely identify it as yours on the CRAY-1A disk. WARNING! This is not a password and it is not written on remote dataset.

"TEXT=........" -- Text parameter is a string of 250 or fewer characters, not necessarily in keyword form. It is of no interest to the CRAY-1A computers (C1 or CA) but is needed by the remote mainframe.

Moving Datasets 3-6 October 1983
EXAMPLE 2: simple DISPOSE of SETUP1 with write password ANDERSON.

DISPOSE, DN=SETUP1, MF=TB, W=ANDERSON.

EXAMPLE 3: ACQUIRE SETUP1 and reDISPOSE under the new name TING, read permission DET and write permission ANDERSON.

ACQUIRE, DN=SETUP1, W=ANDERSON, MF=TB.
DISPOSE, DN=SETUP1, MF=TB, SDN=TING, W=ANDERSON, R=DET.

CDC 7600 Datasets

Recall that CDC 7600-generated datasets are not in CRAY-1A blocked format. A CDC 7600 dataset must be ACQUIRED as an unblocked dataset. Then the ANSI-CODSYL dataset-file label information must be stripped off, CDC 7600 data converted to CRAY-1A format and, finally, the resulting dataset must be re-written in CRAY-1A blocked format. In most cases this entire process can be done automatically during the ACQUIRE process simply by specifying "DF=76" on the ACQUIRE statement. By specifying "DF=76", you cause your dataset to be processed by a utility called TBMCONV, which looks at the label information on the CDC 7600 TEM dataset and decides what type of conversion to apply (e.g. CDC 7600 floating point to CRAY-1A floating point). You may override this default by specifying the "MODE" of conversion yourself. For example if you do not want any conversion, you specify on the ACQUIRE a "DF=M1". A list of the valid mode-conversion numbers and what they mean can be found in the description of TBMCONV in Chapter 4 of this guide.

EXAMPLE: ACQUIRE a CDC 7600-generated dataset named CDC7600, use TBMCONV to convert it to CRAY-1A blocked format, then reDISPOSE it as a dataset named CRAY1A.

ACQUIRE, DN=CDC7600, MF=TB, DF=76.
DISPOSE, DN=CDC7600, SDN=CRAY1A, MF=TB.
For more explanation and examples on how to convert CDC 7600-generated TBM datasets and how to access specific labeled files, see the section in Chapter 4 of this guide on the use of TBMCONV.

**STAGING TAPES**

There are two ways of moving data from tapes to the CRAY-1A computers through the tape drives: (1) linked directly to the CRAY,CA and (2) a link through the IBM 4341. If a tape cannot be read or written through either of these links, it is usually possible to read/write directly on the IBM 4341 and copy to a tape which is readable by the tape drives linked to the CRAY,CA.

Tapes ACQUIRED by the CRAY-1A computers are assumed to be unsegmented and the data on these tapes is in CRAY-1A format. Many times this is not the case. After the tape has been ACQUIRED as a CRAY-1A blocked disk dataset, you must then go through the data and convert it to CRAY-1A format. There are utilities for data format conversion described in Chapter 4, notably DSEG, SCONV and MCONV. Also in the descriptions of these utilities are many examples of solutions to common format problems. After reading this section on how to ACQUIRE your tape as a CRAY-1A blocked disk dataset, you are advised to read Chapter 4 to find a method of converting that dataset to CRAY-1A format data.

**Cray Direct Tape Connection**

This section describes the interface you will have when directly staging tapes to or from the CRAY-1A disks and the tape drives connected to the CRAY,CA. You will be able to use either the ACQUIRE or DISPOSE control statements as described in the previous pages, or the corresponding FORTRAN-callable subroutines. Error messages for this interface may be found in Appendix A of this guide.

**NOTE:** As of the time of this writing, this software is available only on the CRAY,CA machine.

**Terms Used**

The following terms will be used in this section:

- **block** A physical record on a tape.

- **blocked** When this word appears without quotation marks, it is used as IBM uses it in their documentation—that is, referring to a tape with an integral number of logical records per physical record.

- **"blocked"** When this word appears in quotation marks, it is used to denote a dataset in the format to which Cray Research, Inc. refers in their documentation as blocked—that is, the
internal COS format.

byte A string of eight bits. This is one-eighth of a CRAY-1A word.

caracter Synonymous with byte.

dataset A CRAY-1A dataset residing on the disks.

disk mark A logical end-of-file on a disk.

record A logical record. In general, there can be no assumptions made about the relation between logical records ("records") and physical records ("blocks").

tape mark A physical end of file marker written on a tape.

word A string of 64 bits. There are eight characters in a word.

The two simplest control statements to stage a tape are:

\begin{verbatim}
DISPOSE (DN=B12345, MF=MT)
\end{verbatim}

and

\begin{verbatim}
ACQUIRE (DN=V98765, MF=MT)
\end{verbatim}

The DISPOSE statement will copy the user's local dataset called B12345 to a nine-track, half-inch tape of the same name at 6250 bpi.

Correspondingly, the ACQUIRE statement will search the CRAY-1A dataset catalog for the dataset V98765. If the dataset is not found, it will be copied from the user's nine-track, half-inch tape of the same name at 6250 bpi.

COS provides a mechanism for the user dataset name to be different from the tape volume name. For example, the statement:

\begin{verbatim}
DISPOSE (DN=FT10, SDN=B88776, MF=MT)
\end{verbatim}

will take the data which a FORTRAN program wrote to unit 10 and save it on tape B88776.

A Note on Generic Resources

The operating system requires that users declare on their JOB card any generic resource which they will be using within their job, and the mainframe MT falls under this category. To specify this generic resource, include the parameter *MT on your JOB card, as shown below:
The "TEXT" field of the ACQUIRE and DISPOSE statements will be used to allow the user to override the default specification of parameters. The keywords and values allowed in the text field (along with their defaults) are described below. All numbers which the user passes as values associated with these keywords are assumed to be decimal.

**BLKSIZE**
If specified as a decimal integer, this is the maximum size of the physical records to be read or written on the tape in bytes. BLKSIZE must be less than or equal to 1,015,808.

If DF=TR (transparent mode) was specified on the ACQUIRE or DISPOSE statement, the size of the data blocks processed must be a multiple of 4096 bytes (512 words), since that is the size of one disk sector. If there are any blocks which are larger than this value, a warning message will be issued and the data will be truncated. The user-specified value for BLKSIZE will always be rounded upward to a multiple of 4096.

The size of the buffer allocated in CRAY-1A memory is always an integral number of words, so when ACQUIRing data, if BLKSIZE is specified as, for example 4090, it will still be possible to read from the tape a block of 4096 characters, if variable length records are being processed.

BLKSIZE may also be specified as "BLKSIZE=Rn", where n is an integer. This specification can be used to force the tape software to automatically figure out the block size. A temporary buffer of 131,056 bytes (16,382 words) will be allocated and the first n records will be processed using this buffer. After processing the n records, the buffer space will be re-allocated using a value for BLKSIZE which is equal to the largest of the n records.

The default is BLKSIZE=R1 unless DF=TR has been specified. If DF=TR is specified on an ACQUIRE, a value similar to R1 will be used, except that the temporary buffer allocated to process the first record will be 122,880 bytes (15,360 words) long. (NOTE: if DF=CH, the buffers may only be 1/8 as large). This allows the reading of as many as 30 disk sectors (the size of a TBM record) from one tape block. If DF=TR is speci-
fied on a DISPOSE, the default is BLKSIZE=40960 (5,120 words), which will write ten disk sectors on each tape block.

Note that if "BLKSIZE=Rn" is specified (or defaulted), the value to be used for BLKSIZE will be calculated after processing SKIPF or SKIPR. (Descriptions of the SKIPF and SKIPR parameters appear on the following pages).

DEN
Tape density. The supported densities will be DEN=6250, DEN=1600 and DEN=800. The default is DEN=6250.

EOTP
This parameter, which must be specified as EOTP=YES, EOTP=NO or EOTP=EOD, tells whether to allow attempts to process past the reflector at the end of the tape.

The default for DISPOSE requests is EOTP=NO which means that any attempt to process past the reflector will cause a fatal error.

If EOTP=YES is specified, a warning message is issued when the reflector is encountered.

The default for ACQUIRE requests is EOTP=EOD, which means that when the reflector is encountered, all tape reading is stopped, a warning message is issued, an end-of-file and end-of-data are written to the disk file, and the user’s request is considered to be successfully completed. EOTP=EOD is invalid for DISPOSE requests.

ERRMAX
This keyword applies to ACQUIRE requests only, and will be ignored on DISPOSE requests. This is the maximum number of unrecoverable read failures which will be permitted during the reading of the tape. The default is zero if DF=TR is specified. The default is 64 if any other value is specified (or defaulted) for DF. If there are more than this number of blocks which cannot be read in, the user’s request is aborted. NOTE: Deferred implementation.

LABEL
If LABEL=YES is specified, the tape is assumed to have ANSI standard labels. The default is LABEL=NO. NOTE: Deferred implementation.

LRECL
Size of the logical records to be processed, if it is to be different from the physical record size. The default is the value specified, or
defaulted for BLKSIZE so the tape is considered to be "unblocked." **NOTE:** Deferred implementation.

**NFILES**  Maximum number of files to be read from either the tape or the disk. If this parameter is not specified, reading a disk will terminate upon reaching end-of-data and tape reading will stop after a double file mark.

When reading either a tape or disk, each file mark read causes the file counter to be decremented by one.

**NRBCS**  Maximum number of records to be read from either the tape or the disk. An end-of-file is counted as record. When reading the disk, processing is also stopped if an end-of-data is encountered. When reading a tape, a double file mark is ignored--it is treated as an ordinary end-of-file followed by a file with no records in it.

If this parameter is not specified, reading a disk will terminate upon reaching end-of-data, and tape reading will stop after a double file mark. If both NFILES and NRBCS are specified, the indicated number of files will be read in, and then the indicated number of records will be read from the next file.

**NRBCPF**  Maximum number of records to be read from each file on the tape from which the data is being ACQUIRED. This parameter is ignored for DISPOSE requests. As soon as NRBCPF records have been read from a file, the tape is forward spaced to the next file mark. The default for this keyword is infinity, so all records will be processed.

**NTFSBW**  This keyword applies to DISPOSE requests only, and is used to specify the Number of Tape Files Skipped Before Writing. It can be used to append files to a tape. It is our strong recommendation that the user should not attempt multiple DISPOSEs to the same tape, each time incrementing the NTFSBW parameter, as the DISPOSEs will not necessarily be executed in the order they were requested.

**RTRYCNT**  This keyword applies to ACQUIRE requests only, and is ignored on DISPOSE requests. It defines the maximum number of unsuccessful attempts which may be made to read a given record before
that record is flagged as being in error. The default is a very large number. If specified as RTRYCNT=0, no error processing will be done.

**SKIPF**

Number of files to be skipped before starting to read either the tape or the disk. The default is zero. The number specified must be less than 8,388,608.

If both SKIPF and NFILES are specified, SKIPF is processed before starting the counting for NFILES.

When processing a DISPOSE with DF=TR, SKIPF is ignored.

**SKIPR**

Number of records to be skipped before starting to read either the tape or the disk. The default is zero. The number specified must be less than 8,388,608.

When reading either a tape or disk, each file mark read causes the record counter to be decremented by one.

If both SKIPF and SKIPR are specified, the appropriate number of files are skipped first, and then the records are skipped. If both SKIPR and NRBCS are specified, SKIPR is processed before starting the counting for NRBCS.

Thus, the order of processing the parameters is as follows:

1. SKIPF
2. SKIPR
3. NFILES
4. NRBCS

As each counter is being decremented, the next parameter in the list will be ignored until the counter being processed goes to zero.

Note that if "BLKSIZE=Rn" is specified (or defaulted), the value to be used for BLKSIZE will be calculated after processing SKIPF or SKIPR.

**TRANS**

The translation table for character data. This parameter is used when reading or writing character data which is not ASCII characters. This
keyword is ignored unless DF=CH or DF=CB is specified (or defaulted). The default value is TRANS=ASCII, which causes no translation to be done. TRANS=EBCDIC is also supported to allow the reading and writing of EBCDIC tapes to or from ASCII disk files on the CRAY-1A.

**COS-Supported Keywords**

The following is a list of all the parameters to DISPOSE and/or ACQUIRE and what they will mean:

- **DC** (DISPOSE only)
  This parameter is ignored, as it is by the TEM software.

- **DEFER** (DISPOSE only)
  This parameter will function as it is described in CRI documentation.

- **DF** Dataset format. The possible values are as follows:
  - **BB** or **BI** Binary blocked. The dataset on the CRAY-1A disk will be in standard CRAY-1A format. The data contained in the records will be binary data. The records on both the disk and the tape may be of variable length.
  - **CB** or **CH** Character blocked. The dataset on the CRAY-1A will be in standard CRAY-1A format. The data contained in the records will be character data. The records are of variable length.
  - **TR** Transparent. The dataset on the CRAY-1A will be treated as a string of disk sectors, each 4096 bytes long. No assumptions will be made about the contents of the dataset. When reading a tape in transparent mode, the first file mark on the tape always terminates processing. When writing a tape in transparent mode, after the last disk sector has been copied, a double file mark will be written. No file mark will be written at any other time.

  All other possible values for DF are invalid and will cause fatal errors.

- **DN** This parameter will function as it is described in CRI documentation.

Moving Datasets 3-14 October 1983
ED  This parameter will function as it is described in CRI documentation. It is ignored on DISPOSE statements.

ID  This parameter will function as it is described in CRI documentation. It is ignored on DISPOSE statements.

M   This parameter will function as it is described in CRI documentation. It is ignored on DISPOSE statements.

MF  This parameter is required and must be specified as MF=MT.

NOWAIT (DISPOSE only)  This parameter will function as it is described in CRI documentation.

NRLS (DISPOSE only)  This parameter will function as it is described in CRI documentation.

PDN (ACQUIRE only)  If specified, this is the physical label on the tape to be read. The default is the value of DN, as described in CRI documentation.

R   This parameter will function as it is described in CRI documentation. It is ignored on DISPOSE statements.

RT  This parameter will function as it is described in CRI documentation. It is ignored on DISPOSE statements.

SDN (DISPOSE only)  If specified, this is the physical label on the tape to be written. The default is the value of DN, as described in CRI documentation.

SF (DISPOSE only)  This parameter is ignored, as it is by the TBM software.

TEXT  The use of this parameter is described above.

TID  This parameter is ignored, as it is by the TBM software.

UQ  (ACQUIRE only)  This parameter will function as it is described in CRI documentation.
W This parameter will function as it is described in
CRI documentation. It is ignored on DISPOSE
statements.

WAIT (DISPOSE only)
This parameter will function as it is described in
CRI documentation.

If an error occurs while reading a tape block, the
software will retry the operation a few times. If it is
still not possible to read the block, as much data as was
successfully read in will be transferred to disk. The
record will be flagged as being in error, as specified in
Cray-OS Version 1 Reference Manual, SR-0011. At the time
of this writing, it isn't clear which of CRI's library
routines process the error flag which their tape software
sets.

Examples Using
Cray Direct Tape

The following are some examples of things which can be
done:

1. ACQUIRE(DN=B11111, ID=MYDATA, MF=MT, TEXT="BLKSIZE=4096")

A tape consisting of records containing at most,
4096 characters, will be read. The data will be
copied to a CRAY-1A dataset named B11111 which has
an ID of MYDATA associated with it. Each time a
single file mark is found on the tape, an end-of-
file will be written on the disk dataset. When a
double file mark is encountered, an end-of-file fol-
lowed by end-of-data will be written on the disk.

2. DISPOSE(DN=$PLT, SDN=B22222 , MF=NT, DEFERDF=BB,
   TEXT="BLKSIZE=1440, DEN=1600")

The user's blocked dataset $PLT will be copied to a
1600 bpi tape. If there are any records consisting
of more than 1440 characters, a warning message will
be issued and the record will be truncated. Each
time an end-of-file is found in the dataset, a file
mark will be written on the tape. When the end-of-
dataset is reached, a second file mark will be writ-
ten. Since end-of-dataset is always preceded by
end-of-file, this will result in a double file mark
on the tape.

3. DISPOSE(DN=B33333, MF=MT, DF=TR)

The dataset B33333 will be copied to tape in exactly
the format it appears on the disk. B33333 may be a
CRAY-1A "blocked" or "unblocked" dataset. The only
file marks written on the tape will be two file
marks after the last block.
4. DISPOSE (DN=B44444, MF=MT, DF=TR, TEXT='BLKSIZE=4960')

The dataset B44444 will be copied to tape in exactly the format it appears on the disk, except ten disk sectors will be written on each tape block. If the dataset is not a multiple of ten sectors, the last record written will be short. B44444 may be a CRAY-1A "blocked" or "unblocked" dataset.

5. ACQUIRE (DN=B55555, MF=MT, DF=TR, TEXT='BLKSIZE=4096')

This statement may be used to read a tape like the one written in the previous example. The dataset B55555 will be copied to disk in exactly the format it appears on the tape. Ten disk sectors will be written from each tape block.

MOVING DATASETS TO NCAR LOCAL NETWORK NODES

Storing and Retrieving IBM-CMS Files

You may indirectly DISPOSE a Conversational Monitor System (CMS) file from the IBM 4341 to the Terabit Memory System (TBM) for storage. Because the IBM 4341 has no direct link to the TBM, the file transfer must be effected via the CRAY-1A computers.

CMS Text and Program Files

When the NCAR Local Network is used to transfer a CMS file to a CRAY-1A file system, the CMS file becomes an entire CRAY-1A dataset. A CMS end-of-file is equivalent to the Cray end-of-dataset. To specify Cray end-of-file marks within the CMS file to separate different CRAY-1A files on the dataset, insert lines with the four EBCDIC characters "\EDF" between the sections of text representing the different CRAY-1A files. Thus, if you have CMS files MY FILE1 A, MY FILE2 A, and MY FILE3 A, all of which are to be stored on a TBM volume named 'TBMV', you would need to concatenate them together and insert new additional lines as follows:
EXAMPLE 1:

```plaintext
JOB,JN=YOURNAME,US=123412345678,*TB.
ASSIGN,DN=TEMP.
COPYF,I=$IN,O=TEMP,NF=3.
REWIND,DN=TEMP.
DISPOSE,DN=TEMP,SDN=TBMV,MF=TB.
\EOF

{ original text for MY FILE1 A is here }
\EOF

{ original text for MY FILE2 A is here }
\EOF

{ original text for MY FILE3 A is here }
\EOF
```

The above job file would then be sent to the CRAY,C1 with the CLJOB EXEC (or CAJOB if you were sending the job to the CRAY,CA). The three files would be unnamed on the CRAY-1A dataset.

**NOTE:** In the example above, all of the program JCL and the three CMS files are transmitted to the job file on the CRAY-1A (i.e. $IN). There is an automatic character conversion performed from EBCDIC to ASCII when this job file is transmitted to one of the CRAY-1A computers.

To retrieve the volume 'TBMV' created in the above example, you would create a CMS file on the IBM 4341 with any valid Filename and Filetype which contains the program shown below. You would then send this job using the CLJOB (or CAJOB) EXEC: For example, assume the name of the file containing the program below is named JOB DATA:

```
--- -- --- - -- ---LI---·C- -·-L--LII -- --
3-18 October 1983
```
EXAMPLE 2:

CMS File: JOB DATA

JOB,JN=YOURNAME,US=123412345678,*TB.
ACQUIRE,DN=TBMV,MF=TB.
DISPOSE,PDN=TBVM,MF=IO,TEXT=’FLNM=ANY,FLTY=FILE,BOFS=YES’.

To send this job to the CRAY,C1 computer you would type:

CLJOB JOB DATA

Note that the parameter BOFS=YES was specified in the TEXT field of the DISPOSE statement. This parameter forces the NCAR Local Network to replace all Cray volume end-of-file marks with character records containing the four EBCDIC characters "\EOF". Thus, when you read the spool file TBVM FILE onto one of your minidisks, each of your original files will be separated from the next by a single line containing "\EOF". If BOFS=NO is not specified, the BOFS flag defaults to NO, and all of the files stored on the TBM volume will be concatenated into a single CMS file.

Binary CMS Files on the TBM

To ACQUIRE several CMS binary files from the IBM 4341 and then DISPOSE them to the TBM, you would first create a CRAY-1A job which copies those files to a local dataset on one of the CRAY-1A machines. The CRAY-1A job given below will then DISPOSE those same files (now the local dataset) to the TBM.
EXAMPLE 3:

```plaintext
JOB,JN=jobname,US=project/scientist number.
ACQUIRE,DN=file1,MF=IO,DF=BB,TEXT=*FLNM=FILE1,FLTY=DATA*.
COPYF,I=file1,O=IBMBIN.
ACQUIRE,DN=file2,DF=BB,TEXT=*FLNM=FILE2,FLTY=DATA*.
COPYF,I=file2,O=IBMBIN.
ACQUIRE,DN=file3,DF=BB,TEXT=*FLNM=FILE3,FLTY=DATA*.
COPYF,I=file3,O=IBMBIN.
DISPOSE,DN=IBMBIN,DF=TB.
\EOF
```

To reverse the process (e.g. ACQUIRE your IBM binary files from the TBM and send them back to the IBM 4341) you would submit the following CRAY-1A job:

EXAMPLE 4:

```plaintext
JOB,JN=jobname,US=project/scientist number.
ACQUIRE,DN=IBMBIN,DF=TB.
COPYF,I=IBMBIN,O=FILE1.
DISPOSE,DN=file1,DF=BB,TEXT=*FLNM=FILE1,FLTY=DATA*.
COPYF,I=IBMBIN,O=FILE2.
DISPOSE,DN=file2,DF=BB,TEXT=*FLNM=FILE2,FLTY=DATA*.
COPYF,I=IBMBIN,O=FILE3.
DISPOSE,DN=file3,DF=BB,TEXT=*FLNM=FILE3,FLTY=DATA*.
\EOF
```

**NOTE:** You cannot put a binary data file into the job file $\$IN$ because, as was noted above in the first example, everything in your job file is converted from EBCDIC to ASCII and that would be terribly inappropriate for a binary data file. It is for this reason that ACQUIRE statements are used in these latter two examples.
DISPOSING DATASETS TO YOUR RJE SITE VIA RSCS

The IBM 4341 employs a Remote Spooling Control System (RSCS) which may be used to dispose datasets to a remote RJE site. You may ACQUIRE a dataset from the TBM or tape and subsequently DISPOSE it to your remote RJE site which is connected to the IBM 4341 through RSCS. Here is an example of a dataset ACQUIRED from the TBM and subsequently DISPOSED to the RJE printer through RSCS.

ACQUIRE, DN=PRNTFIL, MF=TB.
DISPOSE, DN=PRNTFIL, MF=IO, DC=RS, ^
TEXT='LOC=SITE#, USER=RSCS, DIST="WHO"'.

For detailed information and examples using RSCS, you are directed to the Data Communications guide (in press).

DISPOSING DATASETS TO THE MODCOMP

The mainframe identifier for the MODCOMP is "MC". Remember that the MODCOMP is attached only to CRAY,Cl. A "TID" specifier must be used to specify which remote user is at what remote site. The TID field is 2 digits of site ID followed by 4 digits of sequence number.

The following example works only from the CRAY,Cl machine. It shows how to ACQUIRE a tape and then DISPOSE it to the VAX machine at NCAR's RL-6 site (site #07) with a sequence number 5555.

ACQUIRE, DN=BTAPE, MF=MT.
DISPOSE, DN=BTAPE, MF=MC, TID=075555.

Again you are directed to the Data Communications guide and the section on RJE on the MODCOMP for more information and examples.

MOVING DATASETS TO THE DICOMED

Datasets may be DISPOSEd to the DICOMED graphics processors by using the mainframe specifier "MF=D1". Both plot and print datasets may be sent to the DICOMED. Examples of each type of dataset follow. Additional information may be found in the SCD article "Using the DICOMEDS Online" published in the Selected User Reference Papers of the SCD Graphics manuals.
Plot or Metacode Files from the TBM

To plot a metacode dataset specify a disposition code "DC=PT". This example shows how to plot the dataset MYCODE on 35mm.

```
ACQUIRE, DN=MYCODE, MF=TB.
DISPOSE, DN=MYCODE, MF=DI, DC=PT, DF=BB, TEXT='CAMERA=FILE'.
```

Print Character Data from Tape on Fiche

To "print" a dataset on the DICOMED fiche camera it is necessary only to specify a "DC" of "PR", example of printing CTAPE on fiche.

```
ACQUIRE, DN=CTAPE, MF=MT, DF=CB.
DISPOSE, DN=CTAPE, MF=DI, DC=PR.
```

MOVING DATASETS TO OTHER NODES ON THE NCAR LOCAL NETWORK

In general to move files anywhere on the NCAR Local Network, you must supply the "MF" identifier and specify in the "TEXT" field the remote file specification.

Example: ACQUIRE a file named BIGMAC from the TBM and DISPOSE it to the High Altitude Observatory (HAO) 11/70 machine with pathname /UNIVERSE/NEWKIRK/IT.

```
ACQUIRE, DN=BIGMAC, MF=TB.
DISPOSE, DN=BIGMAC, MF=HA, TEXT='FILE=/UNIVERSE/NEWKIRK/IT'.
```

For more information please see the NCAR manual entitled the NCAR Local Network (TN-181+IA).
CHAPTER 4: INPUT/OUTPUT AND DATASET CONVERSION Routines

INTRODUCTION

In the preceding chapter we talked about moving (staging) datasets between mainframes; the dataset being the largest unit of data transferred between the CRAY-1A computers and any other mainframe. This chapter discusses manipulating the contents of these datasets on the CRAY-1A machines -- a process referred to as "record I/O". This chapter also discusses various supported utilities for converting CDC 7600 data to CRAY-1A format and primitive level utilities for helping you convert data of any machine format.

ROUTINES DESCRIBED

Section I: I/O Routines

1. BRANIO - a direct record access I/O package that emulates the CDC 7600 version. (pg. 4-3)

2. RDTAPE,WRTAPE,IOWAIT - routines for read/writing unsegmented and or unblocked records. (pg. 4-5)

3. UBLOCK, UZBLOK - routines for unblocking fixed length character or binary logical records. (pg. 4-9)

4. RPTIN,RPTOUT - a logical or user-blocked record I/O package. (pg. 4-14)

Section II: Dataset Conversion Routines

1. TBINCNV - Cray procedure for converting CDC 7600 volume to a CRAY-1A dataset. (pg. 4-19)

2. REFRMT - Cray procedure for removing tape segmented data from a TBM volume. (pg. 4-23)

3. SCONV-MCONV - user-called routines for converting CDC 7600 data to CRAY-1A data. (pg. 4-25)

4. DSEG - Cray procedure for removing tape segmentation from tape staged datasets. (pg. 4-29)

5. SBYTES/GBYTES - routines for extracting bits from and placing bits in words. (pg. 4-30)

Section III: Tape Copy Routine

1. TAPBCY - a facility for performing tape copies on the IBM 4341. (pg. 4-33)
SECTION I: I/O

Routine

Dataset record I/O on the CRAY-1A computers can occur only between your program and the CRAY-1A disk subsystem. The reading/writing of records of a dataset on the CRAY-1A computers occurs only between your program residing on one of the CRAY-1A machines and the CRAY-1A disk subsystem. The dataset to be read must have been staged to the disk with an ACQUIRE statement and the dataset written to the disk must then be staged with a DISPOSE statement to another node on the NCAR Local Network for permanent storage.

Cray Research, Inc. (CRI) supports many record level I/O packages, everything from FORTRAN I/O to a FORTRAN callable direct access package called READMS-WRITEMS. The documentation for the CRI-supported packages can be found in FORTRAN (CFT) Reference Manual published by CRI. In addition, NCAR supports BRANIO, RTAPE/WRTAPE/IOWAIT and RPTIN/RPTOUT, the historical I/O packages on the CRAY-1A computers. Everything done in these packages can be done by FORTRAN I/O or by CRI-supported program libraries but we will discuss these packages here because a lot of NCAR software would have to be rewritten if these packages were dropped. If you wish to transport data or programs to another installation, you may find it easier to use these packages as opposed to CRI library routines, because these packages, as well as the data formats they write, are reasonably portable.
This BRANIO package is a partial simulator of the BRANIO software that was previously available on the CDC 7600. At present, this package only performs synchronous FORTRAN reads and writes (i.e., READ(UNIT) list, WRITE(UNIT) list), to the CRAY-1A disks. This package works with CRAY-1A blocked datasets and performs record positioning with the CRAY-1A library routines SETPOS and GETPOS. The software package supports only BRANRD, BRANWT and BRANIO. All of the other entry points that were available on the CDC 7600 are dummies in this package. BRANIO is optional and is used to assign the internal random I/O units as random. You may ignore this entry and provide your own specific ASSIGNs. One other difference from the CDC 7600 version is that the random datasets may be SAVEd or DISPOSEd. The names of these datasets are FT99 and FT98. You must provide the appropriate code for saving or disposing the datasets.

Below is a list of the names and parameters used in the BRANIO entries:

- **NAME**: is the record name which may be any unique set of 64 bits (i.e., one word) as a record name.
- **ARRAY**: is the FORTRAN array from/to which you want to WRITE/READ.
- **LENGTH**: is the number of words to be WRITTEN/READ.

**Entry Points**

- **BRANRD(NAME,ARRAY,LENGTH)**: Reads the random record NAME of length LENGTH into the FORTRAN array ARRAY.

- **BRANWT(NAME,ARRAY,LENGTH)**: Writes the random record NAME of length LENGTH from a FORTRAN array ARRAY.

- **BRANIO**: Assigns the FORTRAN units 98 and 99 as random units. The entry point needs to be called once before BRANRD or BRANWT are called the first time. This call is not need ed if the user calls ASSIGN for those units.

- **BRANCK (NAME)**: Dummy

- **BRANST (NAME)**: Dummy

- **BRANRL (NAME)**: Dummy
LRANIN (NAME) Dummy
LRANOUT (NAME) Dummy
ENTRY POINT ROTAPE

Purpose
To read unsegmented or unblocked records.

FORTRAN Reference
CALL ROTAPE (NUNIT, MODE, NTYPE, NADDR, NWDCNT)

NUNIT A logical tape unit.

MODE† 7-TRACK TAPE:
0 Even parity, no character conversion.
1 Odd parity (binary mode), no character conversion.
2 Even parity (BCD mode), conversion of external BCD to display code (or vice versa if writing a record).

9-TRACK TAPE:
0 Reads ASCII and converts to DPC.
1 Reads ASCII or EBCDIC as binary bit serial.
2 Reads EBCDIC and converts to DPC.

NTYPE FOR TAPE:
0 Used for all tape writes and to read tapes created on the NCAR system. The NWDS from IOWAIT ignores any partial words, and the partial word at the end of the record is not transferred to memory. (1 provides the same options.)
2 Used to read records from another computer. The NWDS from CALL IOWAIT includes a count for any partial word at the end of the record.

Note: Add 4 to any of the above options to ignore read parity errors. The system will not try to correct this error. Thus a type 4 is the same as type 0 but with no re-read on parity errors.

† Character conversion applies only to tapes. Using the disks never involves character conversion on a read or write.
NADDR  Address of the first word in the record, i.e., the first location to be used for storage.

NWDCNT  Number of words to be read or written from the record, or maximum possible record size.

Return Arguments  None.

Printed Messages  ATTEMPT TO USE AN ILLEGAL MODE NUMBER  
                  ATTEMPT TO USE ILLEGAL UNIT NUMBER

Comments  When reading an end-of-file (EOF) with RDTAPE, the first word of the buffer read in for the EOF will contain a BCD 17.

Timing  Data dependent.

ENTRY POINT WRTAPE

Purpose  To write unsegmented or unblocked records.

FORTRAN Reference  CALL WRTAPE (NUNIT,MODE,NTYPE,NADDR,NWDCNT)

NUNIT  A logical tape unit.

MODE†  7-TRACK TAPE:

  0  Even parity (BCD mode), no character conversion.

  1  Odd parity (binary mode), no character conversion.

  2  Even parity (BCD mode), conversion of external BCD to display code (or vice versa if writing a record).

9-TRACK TAPE:

  0  Writes DCP and converts to ASCII.

  1  Writes binary bit serial.

  2  Writes DPC and converts to EBCDIC.

NTYPE  FOR TAPE:

  0  Used for all tape writes and to read tapes created on the NCAR system. The NWDS from

† Character conversion applies only to tapes. Using the disks never involves character conversion on a read or write.

I/O and Conversion  4-6  October 1983
IOWAIT ignores any partial words. (1 provides the same options.)

2 Used to read records from another computer. The NWDS from CALL IOWAIT includes a count for any partial word at the end of the record.

Note: Add 4 to any of the above options to ignore read parity error. The system will not try to correct this error. Thus a type 4 is the same as type 0 but with no reread on parity errors.

NADDR Address of the first word in the record, i.e., the first location to be used for storage.

NWDCNT Number of words to be read or written from the record, or maximum possible record size.

Return Arguments None.

Printed Messages ATTEMPT TO USE AN ILLEGAL MODE NUMBER ATTEMPT TO USE ILLEGAL UNIT NUMBER

Comments None.

Timing Data dependent.

ENTRY POINT IOWAIT

Purpose To wait for the completion of a read or write.

FORTRAN Reference CALL IOWAIT(NUNIT,NSTATE,NWDS)

NUNIT A logical tape unit.

NSTATE Status:

0 good read or write
1 end-of-file (EOF)
2 parity error on read or unable to write
3 end-of-tape (EOT)

NWDS Number of CRAY-1A 64-bit words read or written.

It is possible to call IOWAIT and get a good return before a unit has been used.

Return Arguments None.

Printed Message ATTEMPT TO USE ILLEGAL UNIT NUMBER
Comments

- After a read or write, IOWAIT must be called before reading or writing again from the same unit.

- In a call to RDTAPE or WRTPAE, the array size must be equal to or larger than the NWDCNT specified.

Timing

Data and device dependent.
SUBROUTINE UBLOK (ITAPE, MODE, KTYPE, IBUF, KMAX, KBLOK, ICD, IBOF)

Dimension of Arguments
ICD (KBLOK), IBUF (KMAX)

Latest Revision
December 1982

Purpose
UBLOK unblocks fixed length character or binary logical records (such as card images) from larger physical records on a blocked tape. Each logical record must be an integer multiple of 64-bits long. One or more logical records must be contained in a physical record and logical records must not overlap into the next physical record. See UZBLOK for unblocking records which are a fixed number of bits in length.

Usage
CALL UBLOK (ITAPE, MODE, KTYPE, IBUF, KMAX, KBLOK, ICD, IBOF)

Arguments
On Input
ITAPE
Input dataset logical unit number.

MODE
Mode for tape record.
= 1 BIN, No character conversion.

KTYPE
Not used.

IBUF
Input buffer.

KMAX
Maximum physical record size plus one word.

KBLOK
Number of 64-bit words in each logical record.

On Output
ICD
Output buffer for unblocked logical record
(must be at least KBLOK words long).

IBOF
End-of-file (EOF) flag.
= 0 A logical record has been unblocked.
= 1 Indicates an end-of-file (EOF) was read.

Entry Points
UBLOK
Special Conditions

- Will read from any file on one dataset.
- Logical record must be an integer multiple of 64 bits long.
- Now set to drop physical records with a parity error.
- See UZBLOK for more general case of logical records a fixed number of bits in length.

Common Blocks

NONE

I/O

FORTRAN logical unit ITAPE - input dataset. When an end-of-file is read, the following message is printed:

EOF ON BLOCK READ, PHYS REC = NN,
LOG REC = NN.

When a physical record is longer than KMAX, the following message is printed:

ACTUAL PHYSICAL RECORD = NN,
MAXIMUM PHYSICAL RECORD = NN.

When IOWAIT returns status of 2 OR 3, the following message is printed:

XX TROUBLE ON BLOCK READ TAPE = NN,
STATE = NN, WDS = NN, REC = NN,
LOG REC = NN. THIS RECORD IS NOT PROCESSED.

Precision

SINGLE

Required ULIB Routines

NONE

Specialist

Roy Jenne, NCAR, Boulder, Colorado 80303

Language

FORTRAN

History

Written by Roy Jenne in February 1967.
SUBROUTINE UZBLOK (ITAPE, MODE, KTYPE, IBUF, KMAX, LREP, NWORDS, KSKIP, KBITS, IBOF)

Dimension of Arguments
LREP (NWORDS), IBUF (KMAX)

Latest Revision
December 1982

Purpose
UZBLOK unblocks fixed length character or binary logical records from larger physical records on a blocked tape. The logical records do not have to start at the beginning of the physical record and the logical record may be any number of bits long. If the logical records are an integer number of 64-bit words in length, then UBLOK is more efficient. Logical records may not cross physical record boundaries.

Usage
CALL UZBLOK (ITAPE, MODE, KTYPE, IBUF, KMAX, LREP, NWORDS, KSKIP, KBITS, IBOF)

Arguments
On Input
ITAPE
Input Data Set Logical Unit Number.

MODE
Mode for tape read.
= 1 BIN, No character conversion.

KTYPE
Not Used.

IBUF
Input Buffer.

KMAX
Maximum physical record size plus one.

NWORDS
Number of full 64-bit words to return for each logical record. Must be large enough to contain KBITS.

KBITS
Number of bits in each logical record.

KSKIP
Number of bits to skip at the start of the physical record. The first logical record starts at bit KSKIP+1.
On Output
LREP
Output buffer for unblocked logical record
(must be at least NW)RDS words long).

EOF
End-of-file flag.
= 0 A logical record has been unblocked.
= 1 Indicates an end-of-file (EOF) was read.

Entry Points
UZBLOK

Special Conditions
* Will read files from only one dataset.

* Logical records may be any fixed number of bits in length. If 64*NWORDS .GT. KBITS, the last word of LREP will contain bits not requested. These should be discarded by the user.

* The unblocking from a record stops when there are not enough bits remaining for another logical record.

* Now set to drop physical records with a parity error.

Common Blocks
NONE

I/O
FORTRAN logical unit ITAPE - Input dataset.

When an end-of-file (EOF) or end-of-tape (EOT) is read, the following message is printed:

EOF ON BLOCK READ, PHYS REC = NN,
LOG REC = NN .

When a physical record is longer than MAX, the following message is printed and the program stops:

ACTUAL PHYSICAL RECORD = NN,
MAXIMUM PHYSICAL RECORD = NN .

When IOWAIT returns status of 2 (parity) or 3 (EOT), the following message is printed:

XX TROUBLE ON BLOCK READ TAPE = NN,
STATE = NN, WDS = NN, REC = NN,
LOG REC = NN . THIS RECORD NOT PROCESSED.
<table>
<thead>
<tr>
<th><strong>Precision</strong></th>
<th>Single</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Required ULIB Routines</strong></td>
<td>None</td>
</tr>
<tr>
<td><strong>Specialist</strong></td>
<td>Roy Jenne, NCAR, Boulder, Colorado 80303</td>
</tr>
<tr>
<td><strong>Language</strong></td>
<td>FORTRAN</td>
</tr>
<tr>
<td><strong>History</strong></td>
<td>Written by Roy Jenne in March 1967.</td>
</tr>
</tbody>
</table>
RPTIN, RPTOUT: A Logical Record Manager

Language FORTRAN

Purpose

RPTIN is used to read logical records (reports) from a tape made by RPTOUT (including those made on CDC 7600). RPTOUT is used to pack short, variable-length logical records into larger physical records on tape.

The records are under checksum control. Logical records can be from 1 to 998 words long. The first 16 bits of each logical record must be reserved for use by RPTOUT. Several tapes can be written or read at once, but each unit must have a separate buffer dimensioned in the calling program.

In general, the I/O operations used by the programmer ignore the record blocking. The exceptions to this are:

- A special buffer dimensioned 1006 must be set aside by the programmer for use by RPTIN/RPTOUT in blocking and unblocking the logical records. There must be a separate buffer for each unit in use at the same time.

- There is no backspace command.

- Before putting an end-of-file on a tape, or otherwise terminating writes on a tape, a special RPTOUT command with JL=2 is required (see below).

- After a rewind or before using an NBUF buffer for another tape, NBUF(1) must be set to zero, and the counters in NBUF may be re-initialized if desired.

ENTRY POINT RPTIN

Purpose To read the next logical record from a file written using RPTOUT.

FORTRAN Reference

CALL RPTIN(KUNIT,KBUF,KLOC,KWDS,JJ,KLMAX,JEOF)

Input parameters

- KUNIT Logical unit number (not 5 or 6)
- JJ =1 No longer used, but included to maintain compatibility with existing programs.
- KLMAX Maximum number of words in a report to move to KLOC
Output parameters  

**TBUF**

An array to use for unpacking the records. Dimension KBUF (1006). KBUF(1) must be set to 0 before the first read. RPTIN will read a new physical record when it sees the 0.

KBUF(2-6) May be cleared when the user wishes to reinitialize these counters

KBUF(2) Will have a count of logical records read

KBUF(3) Will have a count of physical records read

KBUF(4) Will have a count of words read from the tape

KBUF(5) Will have the count of bits in the physical record, excluding the checksum

KBUF(6) Will be the same as JEOF

**KLOC**

Array to put report in

**KWDS**

The total number of words in this report

**JEOF**

Status test flag

=0 Good report returned

=1 End-Of-File

=2 Report returned from a record with a bad checksum

=3 End-Of-Tape. The logical record returned with this JEOF is the last on the current tape.

Return Arguments  

None

Printed Messages

RPTIN—BAD CHECKSUM. UNIT, PHYSICAL RECORD, LENGTH, EXPECTED LENGTH, STATUS.

RPTIN—BAD LOGICAL LENGTH. UNIT, LOGICAL RECORD, LENGTH.

RPTIN—BAD PHYSICAL LENGTH. UNIT, PHYSICAL RECORD, LENGTH, EXPECTED LENGTH, STATUS

RPTIN—SHORT PHYSICAL RECORD. UNIT, PHYSICAL RECORD,
LENGTH, EXPECTED LENGTH, STATUS

RPTIN--ATTEMPT TO READ AFTER EOT, DATA MAY HAVE BEEN LOST

THIS IS NOT A RPTIN RECORD. UNIT, PHYSICAL RECORD, LENGTH.

Comments
None

Timing
Data-dependent

ENTRY POINT RPTOUT

Purpose
To write one logical record

FORTRAN Reference
CALL RPOUT (NUNIT, NBUF, LOCRPT, NWDS, JL)

Input Arguments
NUNIT        Logical unit number (not 5 or 6)
LOCRPT       Location of the report (i.e. logical record), for output. The first 16 bits of the report will be used by RPTIN/RPTOUT. Any data contained in the first 16 bits of the first word will be destroyed.
NWDS         Number of words in the report.
JL           Operation indicator

=1           Output this report, assuming a 1006-word NBUF
=2           No report to output; output the reports in NBUF onto the tape. This is used to output the last buffer onto the tape. This also sets NBUF(1) to zero.

Output Arguments
NBUF         An array in which the records are built. Dimension NBUF(1006)

NBUF(1)      Must be 0 before the first write on a tape. RPTOUT will start a record when it sees the 0.
NBUF(2-4)    May be cleared to initialize the counters.
NBUF(2)      Will have a count of logical records output.
NBUF(3)      Will have a count of physical records output.
NBUF(4) Will be a count of the words output. But words in reports = NBUF(4)−2*NBUF(3).

NBUF(5) Not used by RPTOUT.

NBUF(6) Contains the status of the last physical write. May be tested to detect an EOT condition (NBUF(6)=3).

Return Arguments
None

Printed Messages
RPTOUT—BAD PHYSICAL WRITE. UNIT, PHYSICAL RECORD, LENGTH, STATUS

RPTOUT—BAD LOGICAL LENGTH. UNIT, LOGICAL RECORD NUMBER, LENGTH

RPTOUT—ATTEMPT TO WRITE AFTER EOT

Comments
Handling End-Of-Tape (EOT) Conditions with RPTIN/RPTOUT

To avoid confusion because of inconsistencies in end-of-tape handling, it is recommended that tapes not be written up to the end-of-tape mark. If end-of-tape handling is required, the following procedures may be used:

To use RPTOUT to fill tapes completely to the end-of-tape mark, the write status (available in NBUF(6)) must be tested for a value of 3. To continue when end-of-tape is reached, a new tape must be made available and NBUF(6) set to zero. Then writing may continue on the new tape as if the end-of-tape had not been encountered. The logical record written when the end-of-tape condition was detected will appear on the new tape. If you prefer, NBUF(1) through NBUF(6) may be cleared, but then the logical record written when end-of-tape was detected must be rewritten on the new tape.

When using RPTIN to read tapes written as indicated above, a JEOF=3 return indicates the last logical record on the tape.

Layout of a Physical Record Made by RPTOUT

This is a sample layout of a physical record which has only three logical records with lengths of 18, 5, and 32 "64-bit" words respectively. The record then has the following form and detailed format:
Word 1 If the left-most 4 bits contain a 0, this record was written on the CDC 7600. If they contain a right-justified 1 then this record was written on the CRAY-1A. The right-most 60 bits have a right-justified binary count of the words in the physical record. In this example the count is $1 + 18 + 5 + 32 + 1 = 57$.

Word 2 The left-most 12 bits contain a count of 18 in binary. This means that there are 18 words of data in the first logical record. The next 4 bits are reserved for future use.

Word 2-19 The remainder of word 2 (bits 17-64) and all of words 3-19 contain the data in this logical record.

Word 20 Contains a count of 5 in the first 12 bits. Next 4 bits reserved.

Words 20-24 The data in the second logical record.

Word 25 Contains a count of 32 in the first 12 bits. Next 4 bits reserved.

Words 25-56 The data in the third logical record.

Word 57 Is a checksum of the data in words 1-56. This sum is made by the equivalent of an add-and-carry-logical instruction; see the definition of checksum in the glossary.
Recall that if a dataset is written on a machine other than a CRAY-1A, it may be staged onto one of the CRAY-1A machines as a CRAY-1A unblocked dataset but it will likely need to be converted into CRAY-1A format. This section addresses NCAR-supported software for converting "standard" and "non-standard" foreign datasets.

An example of a standard foreign dataset is a TBM volume written by the CDC 7600, while example of a nonstandard dataset would be an 8-bit per word tape written by an aviation facility incremental recorder. We will discuss conversion of standard CDC 7600 TBM volumes first, then standard CDC 7600 written tapes and lastly, some utilities for helping you to convert non-standard datasets.

**ROUTINES FOR CONVERTING CDC 7600 TBM VOLUMES**

**TEMCONV**

In general you don't need to call TEMCONV explicitly for a CDC 7600 TBM volume. When you acquire the volume you specify "DF=76" or "DF=Mx" where (x=0,1,2,...,8,9). We discuss TEMCONV for two reasons: 1) to explain the meaning of the "Mx" or "MODE=" parameter, and 2) to show you how to create an output dataset consisting of specific files from a CDC 7600 TBM volume.

**Purpose**

The utility TEMCONV allows a CRAY-1A job to read TBM volumes written by the CDC 7600 (including volumes disposed from the CRAY-1A to the TBM via the CDC 7600). Given a particular TBM volume, TEMCONV does three things: 1) It unpacks the original records of a desired file or files from the control-word envelope created by the CDC 7600. 2) It converts those records from CDC 7600 format to CRAY-1A format, in a manner determined by a "mode" recorded with the record when it was written; that "mode" may be overridden. 3) It writes the resulting records to a user-specified CRAY-1A dataset.

TEMCONV will not convert all CDC 7600-written volumes to a form which is directly usable on the CRAY-1A computers. Two cases should be noted:

1. If the records written by the CDC 7600 were BUFFERed out and contain mixed-mode data, you may either have to invoke TEMCONV more than once, with different MODE values, to get several different output datasets which may then be selectively read for the portions properly converted. Alternatively, you may have to invoke TEMCONV with "MODE=1", in order to get back the bits written by the CDC 7600, and then convert the resulting records using SCONV and/or MCONV.
2. If the records were written on the CDC 7600 using unformatted FORTRAN writes and the parameter "CONV=LG" was not used on the \*VOLUME card, then either REFORM or DESEG must be used to get rid of CDC 7600 control words and re-create full-length original data records (which may then need to be operated upon by SCONV and/or MCONV, as well).

**JCL**

```
TBMCONV,I=idn,FN=fn1:fn2:...:fn8,MODE=i,O=odn,DIR=ddn,NRLS.
TBMCONV,I=idn,FS=fs1:fs2:...:fs8,MODE=i,O=odn,DIR=ddn,NRLS.
```

I=idn Specifies the dataset from which files are to be converted. This dataset must contain a TEM volume written by the CDC 7600 (acquired from the TBM directly or from a tape written by the CDC 7600 with a "CONV=TB" on the \*VOLUME card). This parameter is required; if it is the only one given, TBMCONV writes on $OUT a directory of all the files found on the volume and quits. See also "NRLS", below.

**FN=...** Specifies files to be converted. "fn1", "fn2", etc., are file names, each up to 17 characters long. Up to eight names may be given, separated from each other by colons. Using FN by itself causes all files to be converted, except for obsolete ones. FN and FS must not both be used in a given invocation of TBMCONV.

**FS=...** Specifies files to be converted. "fs1", "fs2", etc., are file sequence numbers. Up to eight numbers may be given, separated from each other by colons. Using FS by itself causes all files to be converted, including obsolete ones. FN and FS must not both be used in a given invocation of TBMCONV.

**MODE=i** Mode override. Specifying a mode causes TBMCONV to treat all records as if they had been written with that mode. The default is to use the value associated with each record as it was written on the CDC 7600. The possible modes and the actions taken by TBMCONV are as follows:
<table>
<thead>
<tr>
<th>MODE</th>
<th>ACTION OF TBXONV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Converts DPC characters to ASCII, with the last Cray word zero-filled on the right.</td>
</tr>
<tr>
<td>1</td>
<td>No conversion - just transmits the bits of the original record, 64 per word.</td>
</tr>
<tr>
<td>2</td>
<td>Intended to cause conversion from BCD to ASCII. <strong>Not implemented.</strong></td>
</tr>
<tr>
<td>3</td>
<td>No conversion - same effect as 1 - record is assumed to contain ASCII characters.</td>
</tr>
<tr>
<td>4</td>
<td>Intended to cause conversion from EBCDIC to ASCII. <strong>Not implemented.</strong></td>
</tr>
<tr>
<td>5</td>
<td>Converts 60-bit CDC 7600 integers to 64-bit CRAY-1A integers.</td>
</tr>
<tr>
<td>6</td>
<td>Converts 60-bit CDC 7600 reals to 64-bit CRAY-1A reals - if the real has a zero exponent field, it is assumed to be a 60-bit CDC 7600 integer and is converted to a 64-bit CRAY-1A integer.</td>
</tr>
<tr>
<td>7</td>
<td>Converts DPC cards to ASCII cards - like mode 0, except that COSY characters in each record are expanded into sequences of blanks and the result is forced, by truncation or by blank fill, to exactly eighty characters in length. (A few PLIB volumes were COSYed for the sake of efficiency, and this mode was specifically intended for them, but it may be useful to force the output records to be 80-column card images.)</td>
</tr>
<tr>
<td>8</td>
<td>Transparent mode (implies that the original volume was created on the CRAY-1A and disposed with &quot;MP=76,DF=TR&quot;) - either FN or FS may appear, but not with a list.</td>
</tr>
<tr>
<td>9</td>
<td>Transmits the bits of the original record, 60 per word, right-justified, with the four leading bits zeroed.</td>
</tr>
</tbody>
</table>

**odn** Name of the local dataset to receive the converted files. The default name is $TCO$. Separate files in the original volume become separate files on "odn". This dataset is not rewound, either initially or at termination.
DIR=ddn  Directory control. If this parameter appears, a directory of the dataset "idn" is produced on the dataset "ddn". Specifying DIR by itself causes the directory to be written on "SOUT". This dataset is not rewound, either initially or at termination. If "DIR=0" is used, no directory is produced (which is the default).

NRLS  If used, implies that the input dataset is not to be released at termination. (By default, the input dataset is so released.)

General Notes  Binary integers greater than 2 to the 48th power minus 1 in absolute value will not be correctly converted using mode 6: they will be interpreted as floating point numbers instead. This is crucial for persons wishing to use REFRMT, which assumes that a dataset has been converted with MODE=6.

Mode 6 converts CDC 7600 infinites and indefinites into floating point numbers on the CRAY-1A which represent floating point overflows. These overflows are different depending on whether the corresponding CDC 7600 value was a positive or a negative infinite or indefinite, as follows:

<table>
<thead>
<tr>
<th>THE CDC 7600 CONDITION</th>
<th>THE CONVERSION</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive infinite</td>
<td>06000040000000000000003B</td>
</tr>
<tr>
<td>negative infinite</td>
<td>16000040000000000000002B</td>
</tr>
<tr>
<td>positive indefinite</td>
<td>06000040000000000000005B</td>
</tr>
<tr>
<td>negative indefinite</td>
<td>16000040000000000000004B</td>
</tr>
</tbody>
</table>

where the above numbers are expressed in octal.

Example: Reading a CDC 7600 PLIB TBM volume and getting a file named GCM.OLD and putting it on a dataset named GOTIT.

```
ACQUIRE,DN=P04431,MF=TB,
TEMCONV,I=P04431,FN='GCM.OLD',O=GOTIT.
```

Other examples of the use of TEMCONV may be found in Chapter 6 of this guide in the section on examples of ACQUIREing CDC 7600 TBM volumes.
REFRMT: A Utility for Converting Datasets Written in CDC 7600 Binary Tape Format

Many volumes on the TEM were written with FORTRAN statements of the form "WRITE (unit) list" without using the parameter "CONV=LG" on the *VOLUME card for "unit". In such volumes, your logical records are broken into physical records, each of which contains two control words and 510 (or fewer) data words. Such volumes cannot be read directly on the CRAY-1A machines and TEMCONV provides no way of converting them into an easily read form. The utility program REFRMT provides the latter capability (if all the data are integers or reals).

REFRMT is used in the following way: Assume that the volume DJK099 is to be converted and that the user program is to read the data from unit 1. The required JCL is as follows:

```
ACQUIRE,DN=REFRMT,MF=TB.
ACQUIRE,DN=DJK099,MF=TB,DF=M6.
REFRMT,I=DJK099,O=FT01,NA,L=0.
```

The first ACQUIRE obtains the absolute binary program REFRMT from the CRAY-1A disks or from the TEM. The second ACQUIRE obtains the dataset to be converted and does the first part of the conversion, changing CDC 7600 integers and reals to CRAY-1A integers and reals, respectively. Then, the program REFRMT is executed to complete the conversion; the output is written on the dataset FT01. The user program may now be executed with statements of the form "READ (1) list" to read the desired data.

The parameters "I" and "O" must be present to specify the input and output datasets. Neither dataset is rewound (either before the copy or after). File marks encountered are copied. Copying stops only when an "EOD" is found on the input unit or an error occurs.

The parameter "NA" may be omitted if it is desired that REFRMT abort on error conditions (probably the usual situation).

The parameter "L=0" may be omitted; only two or three lines of output are produced by REFRMT anyway. "L=dataset name" may be used to put the print lines on a specified dataset.
Other examples of the use of REFRTM may be found in Chapter 6 in the section on ACQUIREing CDC 7600 TEM volumes.
If the ANSI standard labels have been removed from a CDC 7600 dataset or, if the dataset is an unlabeled tape of CDC 7600 data, it is possible to read the records and convert the data in each record to CRAY-1A formats with a user program. Routines SCONV and MCONV can assist the user in that conversion. Also if the dataset contains segmented tape records, that segmentation can be removed with DSEG.

**SCONV**

**Purpose**
To convert an array containing a CDC 7600 binary bit stream to a corresponding CRAY-1A data array, according to a "mode" value.

**Access**
This routine resides on the CRAY-1A machines and no special control statements are needed to access it.

**Use**
Before invoking SCONV, the dataset to be converted must be on a 1/2-inch magnetic tape read on the CRAY, CA or come from the TBM (as a TBM volume). If the dataset has been ACQUIRED from 1/2-inch tape using MF=I2, nothing more needs to be done to the dataset before running SCONV. If the dataset has been ACQUIRED from the TBM using MF=TB, then it needs to be run through TEMCONV using MODE=1 (or ACQUIRED with DF=M1), to strip off the TBM envelope around the dataset, prior to invoking SCONV on it. Buffer in the bit stream from the source dataset into an array (called SARRAY below), and make a call to SCONV as follows:

```
CALL SCONV(SARRAY,CARRAY,ICDIM,MODE,NBITS,LOCEND,IER)
```

**SARRAY**
Name of the array containing the bit string to be converted from the CDC 7600. For a string NBITS bits long, this array will need to be dimensioned at least (NBITS+63)/64 words.

**CARRAY**
Name of the array to receive the converted values computed from SARRAY. For most MODEs (see below), one storage location is needed in CARRAY for each CDC 7600 word to be converted. For character string modes (MODEs 0 and 3) you will need to reserve 1.25 times the number of CDC 7600 words to be converted.

**ICDIM**
Dimension of the CRAY-1A array CARRAY, used for bounds checking.

**MODE**
Specifies what type of conversion is to take place.
<table>
<thead>
<tr>
<th>Mode</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DPC (all characters converted, i.e., character string)</td>
</tr>
<tr>
<td>1</td>
<td>binary bit serial</td>
</tr>
<tr>
<td>2</td>
<td>BCD—not implemented</td>
</tr>
<tr>
<td>3</td>
<td>ASCII</td>
</tr>
<tr>
<td>4</td>
<td>EBCDIC—not implemented</td>
</tr>
<tr>
<td>5</td>
<td>binary integer (full 60 bits)</td>
</tr>
<tr>
<td>6</td>
<td>floating point and 48-bit integer</td>
</tr>
<tr>
<td>9</td>
<td>60 to 64 bits/no conversion (zeros in top 4 bits)</td>
</tr>
<tr>
<td>10</td>
<td>left-most 8 DPC characters converted to ASCII and put in a single CRAY-1A word (right-most 2 DPC characters lost)</td>
</tr>
<tr>
<td>11</td>
<td>right-most 8 DPC characters converted to ASCII and put in a single CRAY-1A word (left-most 2 DPC characters lost)</td>
</tr>
</tbody>
</table>

**NBITS** Number of bits involved in the conversion.

**LOCEND** Output parameter giving the address (array index) of the last word after conversion.

**IER** Output parameter giving one of several possible error codes.

- 0 successful completion
- 2 illegal mode
- 3 mode currently not implemented
- 4 dimension of CRAY-1A array exceeded
- 6 number of bits (NBITS) not divisible by 6
- 8 number of bits (NBITS) not divisible by 8
- 60 number of bits (NBITS) not divisible by 60
MCONV

Purpose
To perform multiple conversions on mixed-mode records.

Access
This routine resides on the CRAY-1A machines and no special control statements are needed to access it.

Use
Before invoking MCONV, the dataset to be converted must come either from physical tape or from the TBM. If the dataset has been ACQUIRED from physical tape, nothing more needs to be done to the dataset before running MCONV on it. If the dataset has been ACQUIRED from the TBM using MF=TB, then it needs to be run through TBMCONV using MODE=1 (or ACQUIRED with DF=M1), to strip off the TBM envelope around the dataset, prior to invoking MCONV on it. Buffer in the bit stream from the source dataset into an array (called SARRAY below), and make a call to MCONV as follows:

CALL MCONV(SARRAY,CARRAY,ICDIM,ITABLE,LENTAB,IER)

SARRAY
Name of the array containing the mixed-mode record to be converted from the CDC 7600. This array can be dimensioned the same as the array which was written from the CDC 7600 to create the record.

CARRAY
Name of the array to receive the converted values computed from SARRAY. For numeric data, each CDC 7600 integer or floating point number requires one storage location in CARRAY. For character information, you need to reserve 1.25 times the number of array words that were used on the CDC 7600 to store the character information (8 characters per word on the CRAY-1A vs. 10 characters per word on the CDC 7600).

ICDIM
Dimension of the CRAY-1A array CARRAY, used for bounds checking.

ITABLE
Table of conversions to be performed on the record. This is a two-dimensional array where the number of columns of the array is equal to the number of conversions to take place, and for the Ith conversion, ITABLE(1,I) is the mode of the conversion (see discussion of MODE parameter for SCONV in the previous section), ITABLE(2,I) is the number of bits to be used in the conversion (see discussion of NBITS parameter for SCONV in previous section), and ITABLE(3,I) is an output parameter giving the address (array index) of last word after conversion (see discussion of LOCEND parameter for SCONV in previous section).
The number of conversions to be performed.

IER Output parameter giving one of several possible error codes.

- 0 successful completion
- 2 illegal mode
- 3 mode currently not implemented
- 4 dimension of CRAY-1A array exceeded
- 6 number of bits (NBITS) not divisible by 6
- 8 number of bits (NBITS) not divisible by 8
- 60 number of bits (NBITS) not divisible by 60

Notes For character data, if the last CRAY-1A word in the conversion is only partially full, the remainder of the word is filled with blanks. For binary bit serial data (MODE=1), if the last CRAY-1A word in the conversion is only partially full, the remainder of the word is filled with 0.

For DPC conversion (MODE=0), if the number of bits to convert is not a multiple of 6 then the last few bits are truncated. For ASCII conversion (MODE=3), if the number of bits to convert is not a multiple of 8 then the last few bits are truncated.

For integer (MODE=5) or real (MODE=6), if the number of bits to convert is not a multiple of 60 then the last few bits are truncated. EBCDIC (MODE=4) and BCD (MODE=2) conversion are not implemented and if requested will cause the number of bits requested in the conversion to be skipped.

Conversion number K begins with bit N+1 of SARRAY, where N is the sum, for I = 1 to K-1, of ITABLE(2,I). That is to say, each conversion begins with the bit following the bits already converted (or skipped).

Storage of the output from each conversion begins at a CRAY-1A word boundary; the last word of output from a conversion is filled with blanks or zeroes up to a word boundary.

Other examples of the use of SCONV and MCONV may be found in Chapter 6 in the section on the conversion of CDC 7600 to CRAY-1A data.
Purpose
To desegment segmented records written with FORTRAN binary (unformatted) WRITE statements. This routine is the TAPE equivalent of REFRMT. Records written on the CDC 7600 with statements of the form "WRITE(n)..." where n is the logical number of a unit for which the *VOLUME card does not use the CONV=LG option, are written segmented to the disk and subsequently to the TBM, if the volume is descended there. If the volume is descended to physical tape, it will be segmented regardless of whether the CONV=LG option was used or not. DSEG takes such segmented records and rewrites them as unsegmented binary records which can then be buffered in and converted.

Access
To access the binary code for this routine, insert the following statement in your JCL:

ACQUIRE, DN=DSEG, MF=TB.

Use
Before invoking DSEG, the dataset to be desegmented must come either from physical tape or from the TBM. If the dataset has been ACQUIRED from physical tape, nothing more needs to be done to the dataset before running DSEG on it. If the dataset has been ACQUIRED from the TBM using MF=TB, then it needs to be run through TBMCONV using MODE=1 (or ACQUIRED with DF=M1), to strip off the TBM envelope around the dataset, prior to running DSEG on it.

\[ \text{DSEG, I=idn, O=odn.} \]

I=idn Dataset from which files are to be converted. This is a required parameter. DSEG will convert all the files on a dataset.

O=odn Dataset to which the converted files are to be written. This is a required parameter. Examples using DSEG may be found in the section of Chapter 6 on conversion of CDC 7600 segmented tape records.
As was stated earlier your dataset may be in a format for which we do not have conversion software available. We do, however, have utilities that will extract or insert groups of bits from a record disregarding word boundaries on the CRAY-1A. These routines either insert or extract your group of bits in or from a right-justified CRAY-1A word. This allows you to isolate the bit-groupings that you need to convert. These routines are GBYTES/GBYTE and SBYTES/SBYTE.

GBYTES-GBYTE & SBYTES-SBYTE

This is a package of four routines (GBYTES, GBYTE, SBYTES, and SBYTE) that enables you to get bits from and to store bits into a binary record. The bits may be of arbitrary length, up to 64 bits.

Portability Notes

Each of the four routines in this package has (among others) the two arguments IPACK and IOFFS. IPACK is an array containing a binary record and IOFFS is the number of bits to be skipped at the beginning of the record, prior to the first bit transferred. On a particular machine, with a particular word length, it may be tempting to make the array IPACK start in the middle of the binary record and reduce IOFFS accordingly. This temptation should be avoided, as it tends to create code which does not port well.

Byte lengths should be restricted to values less than or equal to the word length of any machine on which the binary records might potentially be used.

ENTRY POINT GBYTES

Purpose

To get bits from a binary record.

FORTRAN Reference

CALL GBYTES (IPACK, IUNPK, IOFFS, NBITS, NSKIP, ITER)

Dimension of Arguments

IPACK(n), IUNPK(NBYTES), where "n" is large enough to hold all the bits of the binary record, at 64 per word.

Usage

The contents of IPACK are treated as a string of 64n bits. The first IOFFS bits are skipped. The next NBITS bits are copied to IUNPK(1), right-justified, with zero fill to the left; the NSKIP bits following it are skipped. Then, NBITS bits are copied to IUNPK(2) and NSKIP bits are skipped, NBITS bits are copied to IUNPK(3) and NSKIP bits are skipped, etc. The process repeats ITER times.

Thus, each IUNPK(I), for I = 1 to ITER, receives the bits of IPACK numbered IOFFS+(I-1)*(NBITS+NSKIP)+1 through
Arguments

IPACK is an array containing a binary record.

IUNPK is an array containing, on output, the bits retrieved from IPACK by the GBYTES call.

IOFFS is an "initial offset", the number of bits to be skipped at the beginning of IPACK. It may have any value greater than or equal to zero.

NBITS is the number of bits to be transferred. Its value must be between 1 and 64, inclusive.

NSKIP is the number of bits of IPACK to be skipped between the bits retrieved.

ITER is the number of times the transfer and strip sequence is to be repeated.

ENTRY POINT GBYTE

Purpose

To get one series of bits from a binary record.

FORTRAN Reference

CALL GBYTE (IPACK, IUNPK, IOFFS, NBITS)

Dimension of Arguments

IPACK(n), where "n" is large enough to hold all the bits of the binary record, at 64 per word.

Usage

The contents of IPACK are treated as a string of 64n bits. The first IOFFS bits are skipped. The next NBITS bits are copied to IUNPK, right-justified, with zero fill to the left. The CALL statement shown is exactly equivalent to:

CALL GBYTES (IPACK, IUNPK, IOFFS, NBITS, 0, 1)

Arguments

IPACK is an array containing a binary record.

IUNPK is an integer variable containing, on output, the bits retrieved from IPACK by the GBYTE call.

IOFFS is an "initial offset", the number of bits to be skipped at the beginning of IPACK. It may have any value greater than or equal to zero.

NBITS is the number of bits to be transferred. Its value must be between 1 and 64, inclusive.
ENTRY POINT SBYTES

Purpose
To store bits in a binary record.

FORTRAN Reference
CALL SBYTES (IPACK, IUNPK, IOFFS, NBITS, NSKIP, ITER)

Dimension of Arguments
IPACK(n), IUNPK(NBYTES), where "n" is large enough to hold all the bits of the binary record, at 64 per word.

Usage
The contents of IPACK are treated as a string of 64n bits. The first IOFFS bits are skipped. The next NBITS bits are replaced by bits from the right end of IUNPK(1); the following NSKIP bits are skipped. Then, NBITS bits are copied from IUNPK(2) and NSKIP bits are skipped, NBITS bits are copied from IUNPK(3) and NSKIP bits are skipped, etc. The process repeats ITER times.

Thus, each IUNPK(I), for I = 1 to ITER, supplies the bits of IPACK numbered IOFFS+(I-1)*(NBITS+NSKIP)+1 through IOFFS+(I-1)*(NBITS+NSKIP)+NBITS. Other bits of IPACK are unaltered.

Arguments
IPACK is an array containing a binary record.

IUNPK is an array containing, on input, the bits to be packed into IPACK by the SBYTES call.

IOFFS is an "initial offset", the number of bits to be skipped at the beginning of IPACK. It may have any value greater than or equal to zero.

NBITS is the number of bits to be transferred. Its value must be between 1 and 64, inclusive.

NSKIP is the number of bits of IPACK to be skipped between each byte stored and the next.

ITER is the number of times the transfer and skip sequence is to be repeated.

ENTRY POINT SBYTE

Purpose
To store one series of bits in a binary record.

FORTRAN Reference
CALL SBYTE (IPACK, IUNPK, IOFFS, NBITS)

Dimension of Arguments
IPACK(n), where "n" is large enough to hold all the bits of the binary record, at 64 per word.

Usage
The contents of IPACK are treated as a string of 64n bits. The first IOFFS bits are skipped. The next NBITS bits are replaced by bits from the right end of IUNPK. The call statement shown above is exactly equivalent to:
CALL SBYTES (IPACK, IUNPK, IOFFS, NBITS, 0, 1)

Arguments

IPACK is an array containing a binary record.

IUNPK is an integer variable containing, on input, the bit to be stored in IPACK by the SBYTE call.

IOFFS is an "initial offset", the number of bits to be skipped at the beginning of IPACK. It may have any value greater than or equal to zero.

NBITS is the number of bits to be transferred. Its value must be between 1 and 64, inclusive.

SECTION III: TAPE COPY ROUTINE ON THE IBM 4341

Tapes written in 7-track format may not be staged to the CRAY-1A computers. They can, however, be read on the IBM 4341 and copied to 9-track format. The NCAR-supported software package "TAPECY" can be used to accomplish this task.

TAPECY

A facility for performing tape copies on the IBM 4341.

Introduction

The tape copy utility on the IBM 4341 is derived from routines used on the CDC 7600. It consists of an EXEC, TAPECY, and FORTRAN routines, TAPECY, USER, and USERF. The program runs on your own virtual machine and no other operations can be performed until the copy is completed. You can disconnect your terminal from the running tape copy and may then make use of the terminal for other applications. To disconnect your terminal, add the command "set CP run" to your PROFILE EXEC file and then enter "disc" after the tape copy program has begun.

The copy is controlled by input lines which must have been previously prepared in a file called "TAPECY DATA A". This file is prepared automatically when using the direct tape copy option "T". The format of the input file is identical to that used on the CDC 7600 with a few extensions appropriate to the IBM 4341. Output is returned to your virtual reader in two parts. One part is identical to that received on the CDC 7600 (the actual program output) and contains information on block and file counts and lengths. Examples of this output are given in later sections. The second part is the IBM 4341 console file generated during the run and should only be of interest when failures are encountered.

Invoking TAPECY

To invoke this procedure, simply enter the command TAPECY and then respond to prompts. For direct tape duplication choose option "T", and prompts for tape labels and densities will be given. This option assumes that (1) an identical copy is desired, (2) the tape terminates on an empty file, and (3) both tapes are 9 track. For more
complex procedures such as combining two tapes onto one or editing records during the copy, the details of the control lines in 'TAPECY DATA A' and user modifiable subroutines will need to be studied. This information is given in the following sections. When you become familiar with these specifications, options which permit direct preparation ("E" or "U") of the input file, "TAPECY DATA A", may be used. Several tape copy jobs may be prepared in one execution of TAPECY, and the jobs will then run sequentially when the begin ("B") key is selected.

**CMS Files Required**

Using the "T" option will create a file named TAPECYxx DATA A on your A disk. The xx suffix to the file name is a sequential number (or blank) which distinguishes the data files in multiple job runs. The "E" or "U" option requires that you create or have already created a file by this same name. All other necessary files reside on the NCARLIBS SUPLIB disk. In order to modify the USER and USERF subroutines as described below, it will be necessary to copy the file USER FORTRAN from the NCARLIBS disk to your A disk, modifying it as desired, and compiling the file before beginning execution of TAPECY.

**Data Control in TAPECY DATA A**

Program control is accomplished by using control lines in the file TAPECYxx DATA A (where xx is a run number when multiple copies are run together). There is one special control line which uses the system MOUNT command to request tape mounts. This is required since the IBM 4341 has no system JCL commands to insure that tapes are mounted before execution begins. The form of the MOUNT command is somewhat different than the other commands.

### MOUNT vsnnum nu nd nr

- **vsnnum** - 6 character tape label
- **nu** - fortran unit number (1 thru 4)
- **nd** - tape density indicator
  - 16=9,800; 17=9,1600; 18=9,6250
- **nr** - read/write ring indicator
  - 0=ring out (read only)
  - 1=ring in (read or write)

Other control cards use a format where the fields are defined as follows:

<table>
<thead>
<tr>
<th>Card Column</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-5</td>
<td>Control word (left justified)</td>
</tr>
</tbody>
</table>
6-10   NREC = number of records or files
21-22   UNIT1 = A logical tape unit (right justified, 1 thru 4)
27-28   UNIT2 = A logical tape unit (right justified, 1 thru 4)
38      Mode of tape on UNIT1
40      Mode of tape on UNIT2
48      Type of tape on UNIT1 (not used)
50      Type of tape to be written on UNIT2 (not used)
57      Error option
        1 = Copy only records without parity errors
        2 = Copy all records

Control Words (column 1-5)

REM      Remark card which contains any remarks in columns 6-80.
REW      Rewind the unit specified in the field UNIT1 (column 21-22).
WEOF     Write end file on the unit specified in the field UNIT1 (column 21-22).
COPY     Copy NREC records (or one file, whichever occurs first) from UNIT1 to UNIT2 with the modes, types, and error option as specified.
CFILE    Copy NREC files as for COPY. The copy will continue until NREC files have been copied or terminated in the USERF routine (NF = 2 or 3). USERF is normally set up to terminate on two consecutive end files.
SKIP     Skip NREC records (or one file, whichever comes first) on UNIT1. The output parameters must be specified even though they are not used.
SFILE    Skip forward NREC file marks (be careful not to run off the end of your tape).
ULOAD    Rewind and unload UNIT1.
Terminates job.

Definitions of Mode and Type

**MODE**
- **0** = ASCII character code conversion.
- **1** = No conversion.
- **2** = EBCDIC character conversion (equivalent to no conversion on the IBM 4341).

**NTYPE**
This option is not used on the IBM 4341.

User Modifiable Subroutines

You may add logic to subroutine USER to edit/select records by examining data in the record. Without additional logic, the record will be copied. USER is called after a record is read, except for a record that is read with a SKIP control card or a record that has a parity error and the error option on the control card has been set to 1.

USERF will copy end-of-files as read on input tape unless logic is added to delete end-of-files. USERF is not called unless the control word on the control card is CFILE and an end-of-file occurs. In the case of two consecutive end-of-files, the second is not copied in the default form of USERF.

If records on the tape to be copied are longer than 8200 32-bit words, then LTH must be set to the longest record length and BUF must be dimensioned by 2*LTH in the main program TAPECY. This will require copying TAPECY FORTRAN to your A disk and compiling after modification.

Any record read with a length of zero will be ignored (not counted as a record read and not processed further). A message will, however, be printed. A record indicated as zero length is usually noise in a record gap.

If an end-of-tape is encountered on a read and the indicated record length is zero, the read will be treated as an end-of-file. If the record length is greater than zero, it will be treated as a record.

If an end-of-tape is encountered on a write, that record will not be copied.

SUBROUTINE USER (IBUF,IBYTES,NFILE,NREC,ISTATE,NN)
All arguments are set by subroutine COPY except NN. USER is called by COPY after each record is read unless:

- End-of-file is read.
- Error option on control card is set to 1 and current record has a parity error.
- Control word is SKIP.

**Input**

- **IBUF** Buffer containing the current record.
- **IBYTES** Length of the current record.
- **NFILE** Number of the current file read under control of the current control card.
- **NREC** Number of records read in the current file.
- **ISTATE** State of the current read.
  - 0 Good read
  - 1 End-of-file
  - 2 Parity error
  - 3 End-of-tape

**Output**

- **NN** A flag set in USER to determine if the record should be copied. It is set to 1 by the unmodified version of USER on NCARLIBS on the IBM 4341. USER must be modified if logic is desired to determine whether a record should be copied.
  - 0 Do not copy
  - 1 Copy

### SUBROUTINE USERF (NRSUM,NRECS,NFILE,NF)

All arguments are set by COPY except NF. USERF is called by COPY only when the control word is CFILE and an end-of-file occurs.

**Input**

- **NRSUM** The total number of records read under control of current control card.
NRBCS  The number of records in the file just completed.

NFILE  The number of the file just completed.

Output

NF  The flag set for action taken in this subroutine. The version on NCARLIBS simply sets NF = 1, except when 2 consecutive end of files are encountered, NF is set to 2. USERF should be modified if logic is desired to determine whether an end of file should be written and/or the next control card should be read. NF options are:

= 0 Continue copy with next file and do not write an end-of-file.
= 1 Same as 0, but write an end of file.
= 2 Go to next control card; do not write an end-of-file.
= 3 Go to next control card after writing an end-of-file.

Examples  EXAMPLE 1:

Suppose we wish to simply copy a binary tape with 5 files and we want 1600 bpi input and 6250 bpi output, then the data file, TAPECY DATA A, would be:

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOUNT input</td>
<td>01 17 00</td>
</tr>
<tr>
<td>MOUNT output</td>
<td>03 18 00</td>
</tr>
<tr>
<td>REW</td>
<td>1</td>
</tr>
<tr>
<td>REW</td>
<td>3</td>
</tr>
<tr>
<td>CFILE</td>
<td>5 1 3 11 00 2</td>
</tr>
<tr>
<td>WEOF</td>
<td>3</td>
</tr>
<tr>
<td>WEOF</td>
<td>3</td>
</tr>
<tr>
<td>DONE</td>
<td></td>
</tr>
</tbody>
</table>

EXAMPLE 2:

If we want to copy to double end-of-file marks and combine two tapes to one, the data file, TAPECY DATA A, might be:

<table>
<thead>
<tr>
<th>Command</th>
<th>Parameter(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOUNT input</td>
<td>01 17 00</td>
</tr>
<tr>
<td>MOUNT output</td>
<td>03 18 00</td>
</tr>
<tr>
<td>REW</td>
<td>1</td>
</tr>
<tr>
<td>REW</td>
<td>3</td>
</tr>
<tr>
<td>CFILE</td>
<td>5 1 3 11 00 2</td>
</tr>
<tr>
<td>WEOF</td>
<td>3</td>
</tr>
<tr>
<td>WEOF</td>
<td>3</td>
</tr>
</tbody>
</table>
EXAMPLE 3:

The example which follows is output from a tape copy which, for reasons of illustration, is unusually complicated and contains a high number of errors.

This procedure will copy two files from the tape assigned on logical unit 1, to the tape assigned on logical unit 3. Then it will copy 50 records, skip 25 records, copy to the next end-of-file and then copy the next file. The print lines beginning with control words are images of the control cards while the other print lines indicate the results of the previous control cards. Note that comments may be used on the control cards to clarify its operation.

The first file had TROUBLE ON A WRITE on record 57. The TROUBLE ON A WRITE message indicates number of record in, number of record out, length in, length out, and status of the write. (A 0 is a good write, 2 is trouble on writing (probably due to a bad spot on the tape, and a 3 indicates and end-of-tape.) The second file had a parity error on record number 16. The first file contained 138 records and the second 97. Since the copy 50 record had the error option set to 1 and encountered a parity error at record number 31, that record was not copied. The first copy 9999 record card was terminated when an end-of-file was encountered after 37 records, each 250 bytes long. The second copy 9999 record card was terminated when an end-of-tape was encountered on the input tape after 40 records. Since the record length was zero, the end-of-tape was treated as an end-of-file. At the end, two end-of-file marks were written on the output tape.
TAPE REQUESTED MOUNT OUTPUT 183 9,6250 RING IN

REW 01
REW 03
CFILE 2 FLS FROM 1 TO 3 MODE 1,1 TYPE 0,0 ERR 2
TROUBLE ON A WRITE 57 57 400 400 2
138 RECORDS READ WITH LENGTHS OF 400 TO 400 BYTES.
138 RECORDS COPIED FILE NO 1 TOTAL READ 138
EOF WRITTEN ON UNIT 3
PARITY ERROR IN RECORD 16
97 RECORDS READ WITH LENGTHS OF 300 TO 400 BYTES.
97 RECORDS COPIED FILE NO 2 TOTAL READ 235
EOF WRITTEN ON UNIT 3
COPY 50 REC FROM 1 TO 3 MODE 1,1 TYPE 0,0 ERR 1
PARITY ERROR IN RECORD 31
50 RECORDS READ WITH LENGTHS OF 400 TO 400 BYTES.
49 RECORDS COPIED FILE NO 0 TOTAL READ 50
SKIP 25 REC FROM 1 TO 3 MODE 1,1 TYPE 0,0 ERR 2
25 RECORDS READ WITH LENGTHS OF 250 TO 250 BYTES.
0 RECORDS COPIED FILE NO 0 TOTAL READ 25
COPY 9999 REC FROM 1 TO 3 MODE 1,1 TYPE 0,0 ERR 2
37 RECORDS READ WITH LENGTHS OF 250 TO 250 BYTES.
37 RECORDS COPIED FILE NO 0 TOTAL READ 37
WEOF 3
COPY 9999 REC FROM 1 TO 3 MODE 1,1 TYPE 0,0 ERR 2
EOF FOUND AFTER RECORD 40, INDICATED RECORD LENGTH= 0
40 RECORDS READ WITH LENGTHS OF 250 TO 250 BYTES.
40 RECORDS COPIED FILE NO 0 TOTAL READ 40
WEOF 3
WEOF 3
DONE

----------------------------------------

I/O and Conversion 4-40 October 1983
Examples of converting 7-track tapes to 9-track can be found in Chapter 6 in the section covering 7-track tape conversions.
CHAPTER 5: EFFECTIVE USE OF THE TBM

INTRODUCTION

This chapter discusses several important aspects of the TBM; the understanding of which will help you enhance your use of the machine. In particular, this chapter will discuss how to conduct operations between your computer jobs and the TBM more efficiently, how to archive datasets on the TBM, and how to use three utilities for maintaining and managing your TBM datasets.

BACKGROUND INFORMATION

The TBM was developed to provide long-term on-line storage of data. The TBM is not a disk-based system for storing and retrieving data; it uses 2-inch magnetic tape which is wound onto reels weighing fifteen pounds each. Each reel holds 43 billion bits which are read and/or written onto the tape reel via an FM-modulated video signal at 5.25 megabits per second. The reels must be manually mounted on one of 12 operational tape transport modules (there are 16 transport modules but only 12 are usually available at any given time).

Given its age, the TBM is a genuine workhorse but it is beginning to suffer from a "Too Many" syndrome: too many users ascending and descending too much data from too many different tapes too often. For example, the TBM on average writes approximately 32 billion bits/day and reads about 89 billion bits/day.

Reading and Writing Your VSNs

The situation with respect to TBM use for tape reads and writes may be summarized as follows:

- SCD has two CRAY-1A computers which must try to use the mass storage resources of only one TBM.
- The number of users on SCD computers has increased from roughly 800 in 1982 to approximately 1200 today. Part of this increase is due to the increased computing capacity gained with the acquisition of the second CRAY-1A. There has NOT been an equivalent increase in mass storage capacity.
- The amount of data being read and/or written by the TBM has skyrocketed. This has necessitated major changes in the way in which Operations manages the data flow as well as the data itself on the TBM.
- The number of VSNs and dedicated tapes has increased. With only 12 tape drives usually operational at any given time and over 700 TBM reels currently user-active, there are larger numbers of tapes which remain off-line and must be physically mounted when a user requests the data on that tape.
At any given time almost 30 tape requests are pending from users. This translates into an immediate backlog of tape mounts and subsequent delays for users in getting their TBM tapes mounted on an available, functioning tape transport. At present, Operations is handling between 120 and 200 TBM tape mounts in a 24-hour period.

This amount of tape activity is degrading the physical components of the TBM hardware for the simple reason that the TBM tape transports were never designed to be used like 1/2-inch magnetic tape drives.

What is Involved in a TBM Tape Mount?

Mounting a fifteen pound TBM tape reel onto a tape transport is very much different than mounting a 1/2-inch magnetic tape onto a tape drive. Every time a tape mount request is received, an operator must take one of the TBM tape transports off-line and remove the reel which is currently on that transport. This reduces the number of on-line transports which are available to other users by one. Next, the operator must determine what tape reel should be mounted and the transport must be aligned to that tape. If both of the alignments are successfully accomplished (and this is not always the case) then the transport is placed back on-line to allow the CRAY-1A computers to access VSNs on that tape. The whole process takes between 5 and 10 minutes per mount assuming everything works properly during the mounting activity.

Dedicated Reels

A dedicated TBM reel is assigned, either wholly or in part, by the mass storage librarian to one user or group of users. Although Operations tries to maintain as many currently used VSNs as possible on-line, a special problem with VSNs on dedicated reels is that they are always off-line until called for by the user. If you need dedicated reels for your work, particularly if you will be generating large amounts of data, SCD Operations must insist that you organize your jobs to use the least possible number of dedicated reels per job. Know where your VSNs are by running the TBMVSN utility (described later in this chapter) with the /T option to list VSNs by source tape. In this way you will usually be able to group your VSN requests by TBM source tape. Other software utilities exist for accomplishing necessary and important VSN maintenance and are described later in this chapter.

Large VSNs

TBM VSNs which are in excess of 500 TBM blocks can cause a variety of problems for you as well as for the TBM. One TBM block is essentially a million bits. Since the error rate for very large VSNs (i.e. 700 - 1,000 blocks) on the TBM is approximately one error in $10^9$ bits, the possibilities of having an error are quite high. A 1,000 block TBM VSN requires just over 3 minutes to read or
write but, because of its size, the data has a good possibility of containing errors. If so, another 3-to-5 minutes are consumed as the TBM attempts to read up a backup copy of the data from an off-line tape. (This requires yet another tape mount).

Associated with every write on the TBM is a read verify and a disperse process. The read verify checks the data just written for readability. The disperse process actually creates a second copy of the VSN. While performing these functions, the log tape on the TBM is unavailable for other writes from both CRAY-1A computers, effectively tying up the system for all users. The larger the VSN which is descended from the CRAY-1A to the TBM, the more likely that the read verify process may fail. Detecting this, the CRAY-1A automatically retransmits the VSN again to the TBM and initiates the process all over again.

Therefore, creating and using very large TBM VSNs is strongly discouraged. Limiting yourself to VSNs between 300 and 400 blocks will provide you with faster TBM acquires and descends, reduced potential for data errors, and enhanced system throughput.

ARCHIVING DATASETS ON THE TBM

General Considerations

Datasets written to the TBM are written to "reels" which hold approximately 5 billion characters each. The number of TBM reel drives available is limited, so if your dataset resides on a reel that isn't mounted there will be a delay until that reel can be mounted by the operator. Anytime you write data to the TBM it is written to the current log reel and then dispersed to the current production reel. Anytime you read a VSN which is on an off-line reel, the system automatically moves that copy of your VSN to the current production reel. For this reason, if you often access the same dataset, it is likely to be "on-line".

VSN Purge Policy

If data written to a TBM reel isn't referenced (or used) within a certain period of time, it is purged from the TBM. This process is termed "ageing off". At present, you have a period of three calendar months within which to reference your TBM dataset and thereby prevent it from being aged off. However, in February of 1984, there will be a new VSN purge policy in effect.

The new VSN purge policy, simply stated, is as follows: On the first Saturday of each month, rather than quarterly, a purge of all VSNs not accessed during the previous three months will take place.
The current quarterly purge of VSNs allows an actual period of up to almost six months to elapse before an unaccessed VSN is purged. The new system will allow a maximum of nearly four months to elapse without access before a VSN is removed from the TBM, thereby reducing the overload of unaccessed VSNs on the TBM by two months.

UNDER THE NEW MONTHLY SYSTEM, YOU WILL NOT NEED TO UPDATE VSNs EVERY MONTH. A quarterly update will save all needed VSNs because the monthly purge will be based upon a three month non-access period as before.

The UPKEEP utility will help you keep track of your VSNs. The use of the ARCHIVE facility is recommended for those VSNs which you may wish to save for long periods without accessing or changing. Please report your VSNs that fit into this category to the mass storage librarian. The librarian will then move such VSNs to an archival dedicated reel.

Exemptions

There are two classes of TBM datasets that are exempt from the monthly purge: 1) those which are on dedicated reels and 2) special SAVE VSNs such as PSTORE volumes.

DEFINITIONS FOR DEDICATED REEL USERS

Log Tape

The TBM reel which receives each dataset from the CRAY-1A machines and from which a copy is dispersed to the production (prod) reel or the designated dedicated reel. The log copy of the dataset or VSN becomes the inactive, backup copy.

Production (prod) Tape

This tape contains the active copy of all VSNs except those on dedicated reels. It becomes an off-line production reel after a certain period of time when compression of the datasets on the reel has been accomplished.

Volume Serial Name (VSN)

Contains six or fewer characters and identifies the volume to be read or written. Starting VSN names with the same two or three letters followed by a number series helps both the user and Operations keep track of them. The use of write passwords prevents overwriting by others.

Dedicated Reel (MVN)

An MVN is assigned in whole or in part to one group of users. An MVN allows the group to keep related VSNs together for most efficient access, or for archival storage. It must be assigned by the mass storage librarian. "MVN" appears on a DISPOSE statement as TEXT="MVN=TL0XXX,...." where "XXX" is an assigned, unique tape number.

Effective Use of TBM 5-4 October 1983
As the name implies, these reels are available to individuals or groups whose data quantity would not fill a whole reel, but whose use pattern makes it sensible to keep VSNs together on one reel.

This is a special read-only dedicated reel for those of you who have "sleeping" VSNs (those VSNs you do not intend to access in the near future but wish to archive for later use without having to update to save them from the quarterly VSN purges). Lists of read-only VSNs to be put on an ARCHIVAL reel may be sent to the mass storage librarian, Room 5, Mesa Laboratory. If you later need to access your archival VSNs a number of times, you should notify the mass storage librarian so that they may be moved to an on-line production reel.

The TBM software moves Volume Serial Names (VSNs) residing on non-dedicated off-line reels to an on-line reel when they are read or written. Thus VSNs written at the same time, but read at different times, eventually become scattered throughout the mass storage library. VSNs on dedicated reels do not migrate to an on-line reel.

**NOTE:** Users of dedicated reels should be aware that disposing a "dedicated" VSN will move it to the current production reel.

The VSN goes first to the log reel, is then dispersed to the dedicated MVN, and the log becomes the inactive copy. This is the default option when writing to a dedicated MVN.

In certain circumstances, it is preferable for Operators to move VSNs to a reel. A form to request such moves is available at the input counter. Off-site users should discuss moves with the mass storage librarian (303-497-1232). For example, if you need to read a large number of volumes it is more efficient to have them on a current on-line reel rather than having them scattered on many different reels, each of which may require tape mounts. You should furnish a current listing by source tape to the librarian, who will arrange the moves. The CRAY-1A utility, TBMVSN, should be used to generate this list, using the /T option. The list should be as current as possible, because Operations' tape compressing activities may move VSNs from one reel to another.

The following statements define the dedicated reel concepts in use in SCD's mass storage system, and provide you with guidelines for their use.

Dedicated reels are assigned by the mass storage librarian for the following purposes:
1. To provide archival storage for datasets (or volumes) which must be kept with few changes for a longer period of time than the quarterly purge allows.

2. To gather on one tape, for ease of access, related datasets which would be scattered through the mass storage library. This may be done on a short or long term basis.

3. Typical dedicated reel users are modelers and/or large data collectors. The decision to assign a dedicated reel is based on SCD Operations efficiency and user need.

Groups assigned dedicated reels agree to follow these guidelines:

1. Dedicated reels will be mounted for "write" purposes during evening, night and weekend shifts only. A message (via MODCOMP, IBM 4341, phone, or over-the-counter) should be given to the operators, asking for a mount and write-align of the specified dedicated reel. If several 1/2-inch tapes will be called for, their names should also be given to the operators. Although dedicated reels are mounted for "write" operations in off-peak hours only, you may run such jobs in the afternoon or early evening and the log copy will wait until the dedicated reel is mounted to disperse an active copy to it.

2. Dedicated reels will be mounted for "read" operations on a second priority basis during very busy periods. At other times, they will join the normal mount queue. Operators need not be given special notification of "read" requests unless the "read" is accompanied by a "write."

3. A VSN naming system for virtual volumes which starts each VSN with the same two or three letters makes life easier for both you and the SCD Operations staff, and is highly recommended.

4. All problems related to the use of dedicated reels may be discussed with the mass storage librarian of the Operations staff at 303-497-1232.

**UTILITIES FOR TBM**

At this writing there are three utilities for updating, maintaining and otherwise keeping track of your TBM datasets. They are UPKEEP, TBMVSN, and DTLIST. UPKEEP is a general maintenance utility while TBMVSN provides you with information about volumes associated with a specific user or project number. DTLIST provides information on
how much writeable space remains on dedicated reels. Since these utilities exist as CRAY-1A procedure datasets residing on the CRAY-1A disks, you must write a CRAY-1A job which calls the desired utility in your JCL statements. These three NCAR-supported utilities and their associated JCL are described below.

UPKEEP

A program named UPKEEP is available on the CRAY-1A computers for processing a file of TBM dataset names. UPKEEP allows you to update valuable volumes to prevent them from being purged and to mark unneeded datasets for deletion. It also enables you to keep a file containing all of the names of your datasets and comments about them for your own records. UPKEEP picks out the names of the datasets from the file, creates a new job file to update or delete each volume as directed, and then executes the new job file.

To use UPKEEP, you would submit a job of the following form:

```
JOB ,JN=jobname ,US=userproject ,*TB.
UPKEEP.
\EOF
{/ input file containing names of TBM datasets }
\EOF
```

The program expects the following format in the input file:

1. A line containing a character in column 1 is assumed to contain a volume name within the first seven characters.

2. If column 1 contains an asterisk ('*'), the following string of 6 or fewer characters is assumed to be a volume name that should be deleted.

3. If the volume is to be deleted and has a read and/or write password(s), these must be supplied. To supply these, the keyword(s) 'W=' and/or 'R=' should be on the same line as the volume name and followed immediately (no intervening blanks!) by the respective password. Passwords longer than 8 characters are truncated to the first eight.

4. If column 1 contains an alphanumeric character, it is taken to be a volume name to be updated. If there are more than six consecutive, non-blank
5. The program continues reading until it reaches an end-of-file mark (\BOR).

**TBMVSN**

A CRAY-1A utility program developed to provide you with TBM Volume Serial Name (VSN) information. The VSN information may be obtained by user and/or project number.

Unless the /T option is used (see Options below), each listed VSN is listed with the following information:

1. Whether the VSN was created by the CDC 7600 or by one of the CRAY-1A computers.

2. Whether or not the VSN has a write password and/or a read password.

3. Whether or not the VSN is exempt from the quarterly purge (by being on a Dedicated or Archival reel or on a special Save list, such as the PLIB or PSTORE volumes).

4. VSNs marked with an asterisk ('*') have errors on the inactive copy. You may wish to ascend and redescend these VSNs.

**Options**

The option turned on by putting a /T after the user or project number lists VSNs by the tape reel containing the active copy, so that users may see how their VSNs are grouped together. This /T information should be used to arrange VSN calls by tape reel, to improve throughput. To save space, the VSNs listed under this option do not have the CDC 7600/CRAY-1A flags, password flags, exempt flags, or error flags. Old VSNs which have no associated user/project number will no longer be listed at the end of the output. To get this list, terminate your data cards with "ENDX" rather than "END". The /D option, when appended to the user/project number, lists VSNs by date of last access.

**Data Cards**

Each data card contains one identifier or user/project number. Identifiers must start in column one. Valid identifiers are: PROJ, USER, END, or ENDX. User or project numbers must start in column five.

A typical deck setup to run the **TBMOVSN** utility is shown below:
DILIST: On-line Dedicated Reel Information

A CRAY-1A utility designed to permit you to determine how much writeable space is left on your dedicated TBM reel. The utility is called DTLIST and may be used to request this information. The output from DTLIST is a numerically ordered listing of all dedicated reels, showing the number of writeable blocks remaining and the percentage of space left on each reel. Reels which are over 90% full are flagged with an asterisk.

A typical deck setup to run the DTLIST utility on the CRAY-1A is illustrated below:

```
JOB,JN=jobname,US=userproject,*TB.
TBMVSN.
EXIT.
\EOF
PROJ
    43310016
    43310016/D
    43310016/T
    43510013/T
USER
    7928
    8039
    8039/D
    8039/T
END (or ENDX)
\EOF
```

Some DOs and DON'Ts for the TBM

- DON'T run jobs on both CRAY-1A computers which require the same VSNs for execution. Reason: When one CRAY-1A is attempting to acquire the needed VSNs the other CRAY-1A must wait for the same VSNs to become available. The system will abort your job.

- DON'T use a WAIT option on a DISPOSE statement, unless it is absolutely necessary. Reason: the WAIT option, unless vital to your needs, holds up a job slot on the CRAY-1A as it waits for the VSN to be descended to the TBM.

- DON'T try to hurry the system! If you are using large quantities of data, please do not submit a second job before the first job is completed. Users
submitting many jobs which output dozens of files at the same time will tie up the data tables on the CRAY-1A computers until the TBM can catch up with the data flow from the CRAY-1A mainframes. This seriously impedes other production runs and reduces CRAY-1A CPU activity to unacceptably low levels.

- DON'T request large numbers of VSNs for one job which are scattered across many different TBM reels. Reason: This requires tape mounts which are directly proportional to the number of TBM reels on which the VSNs are located. Instead, ask the mass storage librarian (303-497-1232) to arrange to move your VSNs to a current on-line production reel. You will need to know the reel numbers where your tapes reside. This information is available by running TBMVSN with the /T option or by looking at the fiche listings in the Consulting Library.

- DON'T create and use TBM VSNs which exceed 300 TBM blocks. Reason: A single TBM block is equivalent to one million bits. The rate of data transmission through the data channels which connect the TBM to the CRAY-1A computers is 5.25 megabits/sec. This rate allows a 300 TBM block VSN to be read from or written to the TBM in just under a minute. During this time, NO OTHER DATA IS BEING PASSED BETWEEN THE TBM AND THE SELECTED CRAY-1A COMPUTER.

- DON'T use the TBM to store data which you do not need. The current overflow of data is one of the most serious problems on the TBM. Keep only the data you need and discard any extraneous VSNs. VSNs may be deleted from the TBM by using the UPKEEP routine. You could, for example, label temporary VSNs in a distinctive way to aid you in deleting them.

- DO make use of the special utilities described above which have been provided for your convenience in keeping track of your VSNs.

- DO label VSNs with a common set of 2 or 3 letters at their beginning. This simple convention will help you—and Operations—keep track of your VSNs.

- DO keep your VSNs small (300 TBM blocks or less). This reduces the chances for errors during a read/write and keeps the time spent in tying up a channel between the CRAY-1A and the TBM to a minimum.

- DO think about what you are trying to do when you submit an ACQUIRE within your CRAY-1A job. Are your
VSNs together on one tape? Are you repeatedly deleting VSNs during a single job requiring multiple ACQUIREs instead of ACCESSes? Can you perform all of your VSN ACQUIREs at the same time? Keeping in mind what you wish to accomplish with the TBM and realizing the consequences of certain activities will make everyone's use of the TBM more productive.

For further assistance, please feel free to contact the SCD Consulting Office (303-497-1278). You may also contact the mass storage librarian (303-497-1232) for help in managing your TBM VSNs.
This chapter provides you with examples of job deck set-ups which may be used to move datasets to and from components of the Mass Storage Subsystem. The examples are grouped into the following sections:

1. Section I: Moving Datasets between the CRAY-1A Disks and the TBM (pg. 6-3).
2. Section II: Moving Datasets between the CRAY-1A Disks and the IBM 4341 (pg. 6-5).
3. Section III: Moving Datasets from CRAY-1A Tape to CRAY-1A Disks (pg. 6-6).
4. Section IV: Moving Datasets from CRAY-1A Disks to CRAY-1A Tape (pg. 6-8).
5. Section V: Setting up Composite Jobs to Stage Datasets (pg. 6-10).
6. Section VI: Conversion of CDC 7600-Generated Data to CRAY-1A Format (pg. 6-12).

* Converting CDC 7600 Generated TBM Volumes
* Converting 9-track CDC 7600 1/2-inch Tape to CRAY-1A Format
* Converting 7-track Tapes to 9-track Tapes

Before illustrating the various methods for moving datasets around on the Mass Storage Subsystem, we review the important components of the command syntax for the ACCESS, DISPOSE, and ACQUIRE statements.

Complete ACQUIRE and DISPOSE parameter descriptions may be found in Chapter 3 of this guide. The ACCESS parameters are a subset of the ACQUIRE parameters. You may wish to refer to this reference list as you work through the examples.

**THE ACCESS STATEMENT:**

```
ACCESS,DN=dn,PDN=pdn,DF=df,ID=uid,ED=ed,R=rd,W=wt,M=mn,JQ,NA.
```

The ACCESS control statement is used to make an existing permanent dataset on the CRAY-1A disks local to a particular job.
THE DISPOSE STATEMENT:

```
DISPOSE, DN=dn, SDN=sdn, DC=dc, DF=df, MF=mf, ID=uid, TID=tid, ED=ed, 
RT=rt, R=rd, W=wt, M=mn, TEXT='text', WAIT, NOWAIT, DEFER, NRLS.
```

The DISPOSE control statement directs a dataset to one of the CRAY-1A computers' output queue for staging to a specified mainframe. The DISPOSE statement also can be used to relinquish access to the named dataset for the job.

THE ACQUIRE STATEMENT:

```
ACQUIRE, DN=dn, PDN=sdn, DF=df, MF=mf, SF=sf, ID=uid, TID=tid, UQ, 
ED=ed, RT=rt, R=rd, W=wt, M=mn, TEXT='text'.
```

The ACQUIRE statement is used to make a dataset permanent and accessible to a job. If the requested dataset is determined to be already residing on the CRAY-1A disks, dataset access is granted to the job. If the dataset is not found on the disks, the request for the dataset is sent to the mainframe specified (or to the mainframe of job origin if no mainframe is specified), which will stage the dataset to one of the CRAY-1A computers. The dataset is then made permanent on the disks and access to it is granted to the job.

In the examples that follow all data movements are performed with ACQUIRE and DISPOSE statements. All machines connected to the CRAY-1A computers through the NCAR Local Network actually pass information to and from the CRAY-1A with NETDISP and NETTAQR which are the two NCAR Local Network equivalents of DISPOSE and ACQUIRE. You may also use the NETTAQR and NETDISP commands. In many cases these are preferred to the Cray ACQUIRE and DISPOSE commands.

**NOTE:** NETTAQR and NETDISP may NOT be used with the TBM.

Documentation on these commands can be found in the NCAR Local Network, which is available from the Documentation Project (Room 17G).
SECTION I: MOVING DATASETS BETWEEN CRAY-1A DISKS AND THE TBM

ACQUIRE a CRAY-1A Dataset from the TBM

A job to ACQUIRE a CRAY-1A format dataset from the TBM and put it on the CRAY-1A disk as a permanent dataset would look as follows:

```
JOB,JN=ANAME,US=1111222222222222,*TB.
ACQUIRE,DN=MYJOB1,MF=TB.
\EOF
```

EXPLANATION:

This CRAY-1A job gets a dataset named MYJOB1 from the TBM. MYJOB1 is already in CRAY-1A format so conversion and format information don't have to be given. The permanent dataset name (PDN) on the CRAY-1A disk will be the same as the local dataset name (DN). The dataset MYJOB1 does not have a unique ID assigned to it, so it will overwrite any other permanent dataset named MYJOB1 on the CRAY-1A disks.

DISPOSE a CRAY-1A Dataset to the TBM

This CRAY-1A job first ACCESSes the permanent dataset MYJOB1 on the CRAY-1A disk and then DISPOSEs it to the TBM.

```
JOB,JN=ANAME,US=1111222222222222,*TB.
ACCESS,DN=MYJOB1.
DISPOSE,DN=MYJOB1,MF=TB.
\EOF
```

EXPLANATION:

The permanent dataset MYJOB1 is ACCESSed for job ANAME and given the local name MYJOB1. MYJOB1 is then DISPOSEd to the TBM and at that point is no longer available to job ANAME. MYJOB1 remains as a permanent dataset on the CRAY-1A disk.
This job setup ACQUIRES a TBM dataset which was written on the now decommissioned CDC 7600 and makes it a permanent dataset on one of the CRAY-1A computers.

```
JOB, JN=ANAME, US=1111222222222222, *TB.
ACQUIRE, DN=MYJOB1, DF=76, MF=TB.
\EOF
```

EXPLANATION:

The specification of DF=76 causes the TBM dataset which is in CDC 7600 format to be converted to CRAY-1A format. If this dataset is pure character or pure nonsegmented binary, then DF=76 will result in the information being converted correctly.
SECTION II: MOVING DATASETS BETWEEN CRAY-1A DISKS AND THE IBM 4341

DISPOSE a CRAY-1A Dataset to the IBM 4341

This job will ACCESS a permanent CRAY-1A dataset and stage it to the IBM 4341.

```
JOB,JN=ANAME,US=1111222222222222.
ACCESS,DN=MYCHAR.
DISPOSE,DN=MYCHAR,MF=IO,DC=ST.
\BOF
```

EXPLANATION:

A permanent character dataset on the CRAY-1A disks named MYCHAR is ACCESSed by job ANAME. The dataset is then DISPOSEd to the IBM 4341 mainframe and will appear in the reader of the virtual machine associated with the user identified by the scientist/project number combination on the job statement. The dataset MYCHAR appears in this virtual machine's reader as MYCHAR FILE. If MYCHAR was instead a binary dataset, the additional parameter 'DF=BB' should be added to the DISPOSE statement.

ACQUIRE a Dataset from the IBM 4341

A job setup to ACQUIRE a dataset from the IBM 4341 would appear as follows:

```
JOB,JN=ANAME,US=1111222222222222.
ACQUIRE,DN=MYJOB1,MF=IO,DF=BB,"TEXT='FLNM=MYJOB1,FLTY=ISRIGHT,USER=MYNAME'."
\BOF
```

EXPLANATION:

This job creates a permanent binary dataset named MYJOB1 on the CRAY-1A disks. It requests that the dataset be staged from the IBM 4341 mainframe, and states that the desired dataset can be found on the 191 disk of the virtual machine belonging to user 'MYNAME' with filename 'MYJOB1' and filetype 'ISRIGHT'.
SECTION III: MOVING DATASETS FROM CRAY-1A TAPE TO CRAY-1A DISK

This section provides you with some working examples of how to read and write CRAY-1A 1/2-inch tapes to or from permanent datasets on the CRAY-1A disks.

Reading a CRAY-1A Tape to a CRAY-1A Disk

The data will be copied to a CRAY-1A dataset named 'B22222' which has an ID of 'MYDATA' associated with it. If there are any records larger than the first record on the tape, no warning message will be issued and those records will be truncated. Each time a single file mark is found on the tape, an end-of-file will be written on the disk dataset. When a double file mark is encountered, an end-of-file followed by end-of-data will be written on the disk.

The job setup to accomplish this task is as follows:

```
JOB,JN=ANAME,US=111122222222,*Mr.
ACQUIRE(DN=B22222,MF=MT,ID=MYDATA)
\EOF
```

Variations on the Theme

Example IIIa: In some instances you may wish to read a tape with physical records not longer than 2048 bytes. The job setup to accomplish this is as follows:

```
JOB,JN=ANAME,US=111122222222,*Mt.
ACQUIRE(DN=B07600,MF=MT,TEXT='BLKSIZE=2048').
\EOF
```

Example IIIb: In another case you may wish to read three files from a CRAY-1A 1/2-inch tape with physical records of 40960 bytes and write a character blocked CRAY-1A dataset with an ID of ANDERSON. To accomplish this, you would submit the following job:

```
JOB,JN=ANAME,US=111122222222,*Mt.
ACQUIRE,DN=B07600,MF=MT,TEXT='BLKSIZE=40960'.
\EOF
```
Example IIIc: This job may be used to read a tape like the one described in example B, above. The dataset 'B55555' will be copied to disk in exactly the format it appears on the tape. Ten disk sectors will be written from each tape block.

Example IIIId: Assume for the sake of this example that tape B66666 is known to contain 5 files. Each file is known to be about 10 blocks long.

This implementation skips the first file on the tape. The first two records on the second file will be ignored. The software will then start copying the data from the tape to the disk. After three file marks have been read from the tape (and the third end-of-file has been written to the disk), 100 more records will be read from the tape and written to the disk. The reading of the tape will then cease. An end-of-file and an end-of-data will be written to the disk.
SECTION IV: MOVING DATASETS FROM CRAY-1A DISK TO CRAY-1A TAPE

The following examples illustrate how to move datasets from CRAY-1A disks to CRAY-1A 1/2-inch tape. In each example the ellipses ( ...... ) represents the job step(s) which access and/or create the dataset being DISPOSEd.

Example IVa: The dataset B33333 will be copied to tape in exactly the format it appears on the disk. B33333 may be a CRAY-1A "blocked" or "unblocked" dataset. The only file marks written on the tape will be two file marks after the last block.

```
JOB,JN=ANAME,US=111122222222,*MT.
           ................
DISPOSE (DN=B33333, MF=MT, DF=TR, TEXT="BLKSIZE=4096")
\EOF
```

Example IVb: The dataset B44444 will be copied to tape in exactly the format it appears on the disk, except ten disk sectors will be written on each tape block. If the dataset is not a multiple of ten sectors, the last record written will be short. B44444 may be a CRAY-1A "blocked" or "unblocked" dataset.

```
JOB,JN=ANAME,US=111122222222,*MT.
           ................
DISPOSE (DN=B44444, MF=MT, DF=TR, TEXT="BLKSIZE=40960")
\EOF
```

Example IVc: Your blocked dataset $PLT will be copied to a 1600 bpi tape. If there are any records consisting of more than 1440 characters, a warning message will be issued and the record will be truncated. Each time an end-of-file (\EOF) is found in the dataset, a file mark will be written on the tape. When the end-of-dataset is reached, a second file mark will be written. Since end-of-dataset is always preceded by end-of-file, this will result in a double file mark on the tape.

Examples 6-8 October 1983
JOB, JN=ANAME, US=111122222222, *MT.

--- JCL to run a job to create $PLT file

DISPOSE (DN=$PLT, SDN=B22222, MF=MT, DF=BB, TEXT='BLKSIZE=1440, DEN=1600')

--- Program to create $PLT

\EOF
In some cases you will be running a job on a mainframe other than the CRAY-1A computers and will require a dataset from magnetic tape, the CRAY-1A disks, or the TBM. It is also possible that you may wish to store the output from such a job on the TBM or magnetic tape. In these instances it is necessary to submit a job to one CRAY-1A to do the staging of the datasets from one mainframe to the other.

You will find that the CRAY-1A job so spawned consists of two job steps constructed out of the first set of examples in this chapter. We will offer some simple examples of such composite jobs in this section without comment. In each example the datasets are assumed to be binary (DF=BB), unless otherwise noted.

**Example Va:** CRAY-1A Tape to CRAY-1A Disk to TBM

```plaintext
JOB,JN=ANAME,US=11122222222,*MT,*TB.
ACQUIRE,DN=B07600,DF=MT,DF=BB.
DISPOSE,DN=B07600,MF=TB,DF=BB.
\EOF
```

**Example Vb:** TBM to CRAY-1A Disks to CRAY-1A 1/2-inch Tape

```plaintext
JOB,JN=ANAME,US=11122222222,*TB,*MT.
ACQUIRE,DN=B07600,DF=TB,DF=BB.
DISPOSE,DN=B07600,DF=MT,DF=BB.
\EOF
```

**Example Vc:** IBM 4341 Disk to CRAY-1A Disk to the TBM

```plaintext
JOB,JN=ANAME,US=11122222222,*TB.
ACQUIRE,DN=MYFILE,DF=IO,DF=BB,
TEXT='USER=MYNAME,FLT=MY,FLTY=FILE'.
DISPOSE,DN=MYFILE,DF=TB,DF=BB.
\EOF
```
Example Vd: TBM to CRAY-1A Disk to IBM 4341 Reader Spool

```
JOB,JN=ANAME,US=11122222222,*TB.
ACQUIRE,DN=MYFILE,MF=TB,DF=BB.
DISPOSE,DN=MYFILE,MF=IO,DC=ST,DF=BB,^ TEXT='USER=MNAME,FLNM=MY,FLTY=FILE'.
\EOF
```

Example Ve: IBM 4341 Disk to CRAY-1A Disk to CRAY-1A 1/2-inch Tape

```
JOB,JN=ANAME,US=11122222222,*MT.
ACQUIRE,DN=MYFILE,MF=IO,DF=BB,^ TEXT='FLNM=MY,FLTY=FILE,USER=ME'.
DISPOSE,DN=MYFILE,SDN=XXXXXX,MF=MT,DF=BB.
\EOF
```

Example Vf: CRAY-1A 1/2-inch Tape to CRAY-1A Disk to IBM 4341 Reader Spool

```
JOB,JN=ANAME,US=11122222222,*MT.
ACQUIRE,DN=MYFILE,MF=MT,DF=BB.
DISPOSE,DN=MYFILE,MF=IO,DC=ST,DF=BB,^ TEXT='FLNM=MY,FLTY=FILE,USER=ME'.
\EOF
```
SECTION VI: CONVERSION OF CDC 7600-GENERATED DATA TO CRAY-1A FORMAT

CDC 7600-generated data, based on a 60 bit word with different integer and floating point formats and with a different character representation, must be converted to corresponding CRAY-1A format before that data can be used on one of the CRAY-1A computers. The following examples show how to convert CDC 7600 TBD volumes and CDC 7600 tapes written in a variety of CDC 7600 formats to CRAY-1A datasets with corresponding formats.

Example VIa: To convert a CDC 7600 TBD volume containing integer, floating point or mixed records to a CRAY-1A dataset. The dataset was written using FORTRAN unformatted binary writes.

Creation on CDC 7600

CDC 7600 JCL:

*VOLUME,11,VSN=xxxxxx,STAGEIN=ZS,STAGEOUT=MD

CDC 7600 FORTRAN:

PARAMETER (N=5000)
INTEGER X(N)
REAL Y(N)

C
C INTEGER RECORDS
C
WRITE (11) (X(I),I=1,N)
WRITE (11,MODE=1) (X(I),I=1,N)
WRITE (11,MODE=5) (X(I),I=1,N)

C
C REAL RECORDS
C
WRITE (11) (Y(I),I=1,N)
WRITE (11,MODE=1) (Y(I),I=1,N)
WRITE (11,MODE=6) (Y(I),I=1,N)

C
C MIXED INTEGER AND REAL RECORDS
C
WRITE (11) (X(I),Y(I),I=1,N)
WRITE (11,MODE=1) (X(I),Y(I),I=1,N)
CRAY-1A Conversion

CRAY-1A JCL:

ACQUIRE, DN=REFRT, MF=TB.
ACQUIRE, DN=xxxxxx, MF=TB.
TBMCONV, I=xxxxxx, FS=1, MODE=6.
REWIND, DN=$TCO.
REFRMIT, I=$TCO, O=FT11.
REWIND, DN=FT11.
CFT.
LDR.
DISPOSE, DN=FT11, SDN=xxxxxx, MF=TB.

CRAY-1A FORTRAN:

PARAMETER (N=5000)
INTEGER X(N)
REAL Y(N)

C INTEGER RECORDS
C READ(11) (X(I), I=1,N)
C REAL RECORDS
C READ(11) (Y(I), I=1,N)
C MIXED INTEGER AND REAL RECORDS
C READ(11) (X(I), Y(I), I=1,N)

October 1983
Examples
Notes: 1. MODE 6 works on mixed floating point and integer records.

2. Note that the CDC 7600 volume was descended with default conversion CONV=BN. The records will be segmented into 510 word records with 2 control words per record. This requires use of REFRMT on the CRAY-1A.

3. REFRMT does not rewind its input data set.

4. A user could define a MODE parameter for a FORTRAN write statement on the CDC 7600. For binary writes, as above, the default mode is MODE=1, signifying binary bit serial. For binary writes, the user could alternatively specify MODE=5 for writing integers, and MODE=6 for writing floating point numbers. In both cases, one can use MODE=6 on the TBMOONV statement.
Example VIb: Conversion of a CDC 7600 TEM dataset containing mixed character and numeric records to a CRAY-lA dataset. Records were written using FORTRAN unformatted binary write statements.

Creation on CDC 7600

CDC 7600 JCL:

*VOLUME, 11, VSN=xxxxxx, STAGEIN=ZS, STAGEOUT=MD

CDC 7600 FORTRAN:

PARAMETER (N=2000)
COMMON X(N), Y(N), Z(N)
INTEGER X, Z
REAL Y

DO 10 I=1, N
  X(I)=I
  Y(I)=FLOAT(I)
  Z(I)=10HABCDEFGHIJ
10 CONTINUE

C
C MIXED RECORDS
C
WRITE(11) X, Y, Z

October 1983 6-15 Examples
CRAY-1A Conversion

CRAY-1A JCL:

ACQUIRE, DN=xxxxxx, MF=TB.
ACQUIRE, DN=DSEG, MF=TB.
TBMCNV, I=xxxxxx, FS=1, MODE=1.
DSEG, I=$TCO, O=FT11.
REWIND, DN=FT11.
CFT.
LDR.
DISPOSE, DN=FT11, SDN=xxxxxx, MF=TB.

CRAY-1A FORTRAN:

PARAMETER (N=6000, M=6500, LENTAB=3)
INTEGER SARRAY(N), CARRAY(M), TABLE(3, LENTAB), IER
COMMON X(2000), Y(2000), Z(2500)
INTEGER X, Z
REAL Y
EQUIVALENCE (X, CARRAY)
EQUIVALENCE (Y, CARRAY(2001))
EQUIVALENCE (Z, CARRAY(4001))

C
C MIXED RECORDS, WRITTEN BINARY WITH FORTRAN BINARY WRITE
C
READ(11) X, Y, Z

C
C TWO THOUSAND WORDS OF INTEGER
C
TABLE(1,1) = 5
TABLE(2,1) = 60*2000

C
C TWO THOUSAND WORDS OF FLOATING POINT NUMBERS
C
TABLE(1,2) = 6
TABLE(2,2) = 60*2000

C
C TWO THOUSAND WORDS OF DPC CHARACTER
C
TABLE(1,3) = 0
TABLE(2,3) = 60*2000

C
CALL MCONV(SARRAY, CARRAY, M, TABLE, LENTAB, IER)
Notes

1. The array SARRAY used to buffer in the CDC 7600 record was dimensioned the same as the array block used to write the records on the CDC 7600. This is convenient, although only 15 words on the CRAY-1A are needed for every 16 words on the CDC 7600 for binary bit serial. The BUFFER IN will of course stop at the end of the record.

2. Each CDC 7600 integer or floating point number written requires one CRAY-1A storage location. The number of words required for character information is 1.25 times that required for the CDC 7600.

3. MCONV will not attempt a conversion which would otherwise overflow the converted array, CARRAY, above.

4. DSEG is used to desegment the data as opposed to REFMT, due to the presence of character data. TBMCONV strips the TBM envelope off.

Example VIc: To convert a CDC 7600 dataset containing mixed character and numeric records, written with BUFFER OUT or WRITAPE statements.

CDC 7600 JCL:

*VOLUME, 11, VSN=-xxxxxx, STAGEIN=ZS, STAGEOUT=MD

CDC 7600 FORTRAN:

PARAMETER (N=2000)
COMMON X(N),Y(N),Z(N)
INTEGER X,Z
REAL Y

DO 10 I=1,N
   X(I)=I
   Y(I)=FLOAT(I)
   Z(I)=10HABCDEFGHIJ
10 CONTINUE
C
C MIXED RECORDS
C
CALL WRITAPE(11,1,0,X,3*N)
BUFFER OUT(11,1) (X(I),Z(N))
BUFFER OUT(11,0) (X(I),Z(N))
CRAY-1A Conversion

CRAY-1A JCL:

ACQUIRE, DN=xxxxxx, MF=TB.
TB4CONV, I=xxxxxx, FS=1, O=FT11, MODE=1.
REWIND, DN=FT11.
CFT.
LDR.

CRAY-1A FORTRAN:

PARAMETER (N=6000, M=6500, LENTAB=3)
INTEGER SARRAY(N), CARRAY(M), TABLE(3, LENTAB), IER
COMMON X(2000), Y(2000), Z(2500)
INTEGER X, Z
REAL Y
EQUIVALENCE (X, CARRAY)
EQUIVALENCE (Y, CARRAY(2001))
EQUIVALENCE (Z, CARRAY(4001))

C C MIXED RECORDS, WRITTEN BINARY WITH BUFFER OUT OR WRTAPE
C
CALL RDTAPE(11, 1, 0, X, 3*N)
CALL IOWAIT(11, NSTATE, N)
-or-
BUFFER IN(11, 0) (SARRAY(1), SARRAY(N))
IF (UNIT(11)) 99, 1000, 2000

C C TWO THOUSAND WORDS OF INTEGER
C
TABLE(1, 1) = 5
TABLE(2, 1) = 60*2000

C C TWO THOUSAND WORDS OF FLOATING POINT NUMBERS
C
TABLE(1, 2) = 6
TABLE(2, 2) = 60*2000

C C TWO THOUSAND WORDS OF DPC CHARACTER
C
TABLE(1, 3) = 0
TABLE(2, 3) = 60*2000

C CALL MOONV(SARRAY, CARRAY, M, TABLE, LENTAB, IER)
Notes

1. The array SARRAY used to buffer in the CDC 7600 record was dimensioned the same as the array block used to write the records on the CDC 7600. This is convenient, although only 15 words on the CRAY-1A are needed for every 16 words on the CDC 7600 for binary bit serial. The BUFFER IN will of course stop at the end of the record.

2. Each CDC 7600 integer or floating point number written requires one CRAY-1A storage location. The number of words required for character information is 1.25 times that required for the CDC 7600.

3. MCONV will not attempt a conversion which would otherwise overflow the converted array, CARRAY, above.

4. TBMCONV strips the TBM envelope off.

Example VId: To convert a 9-track tape copy of a TBM volume, created on the CDC 7600 and containing mixed floating point and integer data.

Creation on the CDC 7600

CDC 7600 JCL:

*VOLUME,11,VSN=xxxxxx,STAGEIN=MA,STAGEOUT=DT,CONV=TB.

CRAY-1A Conversion

CRAY JCL:

ACQUIRE,DN=xxxxxx,DF=MT,TEXT='BLKSIZE=122880'.
TBMCONV,I=xxxxxx,O=yyyyyy,MODE=6.
DISPOSE,DN=yyyyyy,DF=TB.

Example VId: To convert a 9-track tape, written on the CDC 7600 as a 1600 bpi binary bit serial tape containing integer, floating point or character data, written using BUFFER OUT or WRTAPE statements.
CDC 7600 JCL:

*VOLUME, 10, VSN=xxxxxx, STAGEOUT=DT, TAPE=16, 9, CONV=BN

-or-

*ASSIGN, xxxxxx=10, 9, 16

*TLIB, 10, DT, BN

-or-

*ASSIGN, xxxxxx=10, 9, 16

CDC 7600 FORTRAN:

BUFFER OUT(10, 1) (IARRAY(1), IARRAY(N))

IF( UNIT, 10 ) ........

-or-

CALL WRTAPE(10, 1, 0, IARRAY, N)

CALL IOWAIT(10, NSTATE, NWDS)

CRAY-1A JCL:

ACQUIRE, DN=xxxxxx, MF=MT, DF=BB, TEXT=’DEN=1600’.

ASSIGN, DN=xxxxxx, A=FT10.

CRAY-1A FORTRAN:

PARAMETER (N=1000, M= N*1.25)

NSIZE = N

MODE = 5

BUFFER IN(10, 1) (IARRAY(1), IARRAY(N))

IF( UNIT(10) ) .......

-or-

CALL RDTAPE(10, 1, 0, IARRAY, N)

CALL IOWAIT(10, NSTATE, NWDS)

CALL SCONV(IARRAY, ICARAY, NSIZE, MODE, N*60, LOCEND, IER)

IF( IER .NE. 0 ) ...

Examples 6-20

October 1983
Notes

1. For integer or floating point conversion, the number of 64-bit CRAY-1A words is equal to the number of 60-bit CDC 7600 words and thus may be dimensioned by the same value as given in your CDC 7600 program, represented above by N.

   For character conversion the CRAY-1A array must be at least 25% longer than the equivalent array on the CDC 7600, represented by M.

2. The input statement (BUFFER IN or RDTAPE) will actually attempt to read more words than are on the record; this will result in the entire record being read and does not constitute an error.

3. Note the difference in the statement structure of IF(UNIT) test between the CDC 7600 and CRAY-1A.

4. MODE=5 conversion will correctly convert all CDC 7600 integers; both the 48-bit integer (maximum size of integer for arithmetic operations) and 60-bit integer (maximum expressible integer on CDC 7600). If only 48-bit integers are expected in the record then the MODE=6 conversion (floating point) will correctly convert them.

5. Floating point conversion is nearly identical to integer conversion: the only difference is that MODE=6 is substituted for MODE=5 in the call to routine SCONV. Indefinite, infinite, and unnormalized values on the CDC 7600 (exponents of 1777, 3777, 4000 octal respectively) will yield an infinite value in the CRAY-1A representation (an exponent of 60000 octal).

6. DPC to ASCII character conversion is also similar to integer conversion: MODE=3 is substituted for MODE=5 and NSIZE=M (which is 125% of N) is substituted for NSIZE=N as the CRAY-1A array dimension in the call to routine SCONV.

7. 800 bpi density runs identically to the 1600 bpi case above, with the appropriate changes in CDC 7600 and CRAY-1A JCL.

8. If the tape had been written as a segmented tape (using unformatted FORTRAN write statements on the CDC 7600 as opposed to the BUFFER OUT/WRTAPE statements shown), it could be read by essentially the same job with the addition of the following lines to the CRAY-1A JCL:
These lines should replace the ASSIGN statement in the example.

9. Currently, the maximum size of a record being read through the MF=MT, is less than or equal to 1,015,808 bytes.

<table>
<thead>
<tr>
<th>ACQUIRE, DN=DSEG, MF=TB.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSEG, I=xxxxxx, O=yyyyyy.</td>
</tr>
<tr>
<td>REWIND, DN=yyyyyy.</td>
</tr>
<tr>
<td>ASSIGN, DN=yyyyyy, A=FT10.</td>
</tr>
</tbody>
</table>
CONVERTING 7-TRACK TAPE TO 9-TRACK TAPE ON THE IBM 4341

Introduction
The general task of converting 7-track tapes to the CRAY-1A machines is quite similar to 9-track conversion with an additional step of first converting the 7-track to 9-track tape. The reason this step is necessary is that there are no 7-track tape drives on the CRAY-1A machines. Hence, the first step to accomplish is to copy the 7-track tape to a 9-track tape drive on the IBM 4341. Step two is to read the resulting 9-track tape on the CRAY,CA computer just as other CDC 7600-generated tapes have been read and described in the previous examples. The tape copy process is done on the IBM 4341 using the TAPECY command from NCARLIBS. A complete description of this EXEC is available on-line on the IBM 4341 through the HELP facility.

At this writing, only the CRAY,CA machine has the capability of reading 9-track tapes. CRAY,CI implementation of this feature is deferred at this time.

PLEASE NOTE: There are many possible problems in converting a 7-track tape to a 9-track tape. The example shown will work correctly in about half of the cases. Errors will occur if the number of bits in the record is not a multiple of 8 bits. You will find extraneous bits appended to your records. Be advised that the process is not guaranteed to be straightforward, and there is a good possibility you will need to work with one of the consultants to complete the conversion.

When using TAPECY most of its prompts are self-explanatory. However, there are a few things that you should know before using it for 7-track to 9-track conversion. You will be asked (among other things) for the input tape density. The prompt also includes three of the most typically used densities, all of which pertain to 9-track tape. You will use none of these, but instead will enter one of the modes listed below depending on the tape density and its parity:

<table>
<thead>
<tr>
<th>MODE</th>
<th>7 TRACK DENSITY</th>
<th>PARITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>200</td>
<td>odd</td>
</tr>
<tr>
<td>6</td>
<td>556</td>
<td>odd</td>
</tr>
<tr>
<td>11</td>
<td>800</td>
<td>odd</td>
</tr>
<tr>
<td>5</td>
<td>200</td>
<td>even</td>
</tr>
<tr>
<td>10</td>
<td>556</td>
<td>even</td>
</tr>
<tr>
<td>15</td>
<td>800</td>
<td>even</td>
</tr>
</tbody>
</table>
Note: be very careful in the selection of the mode for the input tape. If the wrong mode is selected, the tape will often run away. This is because the tapes don’t stop at an end-of-tape mark, forcing the operator to manually reload the tape.

The TAPECY EXEC will also prompt you for the output tape density. We recommend using a value of 17 which represents 9-track 1600 bpi; since this is the default value, you may simply press return to get that value.

When even parity tapes are converted to 9-track a character translation occurs from (the assumed) BCD characters into EBCDIC characters producing a 9-track odd parity tape in EBCDIC characters. (In your CRAY-1A job, the ACQUIRE statement will use the DF=CB keyword which will cause conversion of the EBCDIC characters into ASCII characters before the dataset is sent to the CRAY-1A.)

Odd parity 7-track tapes are taken in bit serial form and written to 9-track with no conversion. (In your CRAY-1A job, the ACQUIRE statement will use the DF=-BB keyword which will cause no conversion before the dataset is sent to one of the CRAY-1A computers.)

TAPECY will perform all the tape mounts, and will detach the drives when the job has been completed. This will not be the case if TAPECY terminates abnormally. Abnormal termination will cause one or more error messages to be sent to your terminal. In that event you must detach the drives manually, or it will severely affect Operations.

The message MOUNT ACCEPTED means that the tape is in the mount queue; the message TAPE xxx ATTACHED will appear once the tape has been mounted. (Where ‘xxx’ represents a virtual address, eg. 181) Often times the confusing message xxx DETACHED appears on the user screen; it has no meaning and should be ignored. In particular do not send the operator a query message since chances are that he/she hasn’t seen anything on the system console yet.

One further note should be made. In your CRAY-1A job, when using the ACQUIRE,DN=xxxxxxx,MF=I2, only two dataset formats will be used in reading the converted 9-track tapes. DF=CB is used for tapes that have been copied from 7-track even parity tapes, while DF=-BB will be used for tapes that have been copied from 7-track odd parity tapes. The only other legal value for DF is TR and is not used in any tape conversions. Any other value could have serious consequences and should not be used.

Examples 6-24 October 1983
EXAMPLE: Conversion of a 7-track 800 bpi, binary bit serial tape having odd parity, created on the CDC 7600 using BUFFER OUT or WRTAPE statements, containing data of mixed types.

Creation on CDC 7600

CDC 7600 JCL:

*VOLUME,10,VSN=xxxxxx,STAGEDOUT=DT,TAPE=8,7,CONV=BN
-or-
*ASSIGN,xxxxxx=10,7,8
*TLIB,10,DT,BN
-or-
*ASSIGN,xxxxxx=10,7,8

CDC 7600 FORTRAN:

BUFFER OUT (10,1) (IARRAY(1),IARRAY(N) )
IF( UNIT, 10 ) ........
-or-
CALL WRTAPE (10,1,0,IARRAY,N)
CALL IOWAIT (10,NSTATE,NWDS)

CONVERSION: The conversion process will take two major steps:

1. Conversion from a 7-track tape to a 9-track tape using the TAPECY EXEC on the IBM 4341.

2. Conversion from CDC 7600 format data to CRAY-1A format data using the MCONV utility on the CRAY-1A.

IBM Conversion (7-track to 9-track)

CMS Command:

TAPECY

TAPECY Interactive Session:
## TAPECY Interactive Session: (continued)

<table>
<thead>
<tr>
<th>PROMPT</th>
<th>USER RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>This procedure allows... (RETURN TO CONTINUE)</td>
<td>&lt;return&gt;</td>
</tr>
</tbody>
</table>

Choose one of the following:
- Use TAPECYL DATA as is (U).
- Edit TAPECYL DATA (E).
- Enter tape VSN’s for one-to-one tape copy (T).
- Abort (A).

T <return>

---

<table>
<thead>
<tr>
<th>PROMPT</th>
<th>USER RESPONSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter input tape number.</td>
<td>xxxxxx &lt;return&gt;</td>
</tr>
<tr>
<td>Choose input density.</td>
<td>11 &lt;return&gt;</td>
</tr>
<tr>
<td>Enter output tape number.</td>
<td>yyyyyy &lt;return&gt;</td>
</tr>
<tr>
<td>Choose output density.</td>
<td>&lt;return&gt;</td>
</tr>
</tbody>
</table>

Choose one of the following:
- Use TAPECYL DATA as is (U).
- Edit TAPECYL DATA (E).
- Enter tape VSN’s for one-to-one tape copy (T).
- Abort (A).

OR type "B" to begin execution of 1 tape copies. | B <return> |

### Notes

- Note 6: Additional notes or instructions related to the process.
At this point we have a 9-track copy of the 7-track tape. We now proceed as we would with any CDC 7600-generated 9-track tape.

**Cray-1A Conversion**

**Cray-1A JCL:**

```
ACQUIRE, DN=yyyyyy, MP=MT, DF=BB, TEXT='DEN=1600'.
ASSIGN, DN=yyyyyy, A=FT10.
```

**Cray-1A FORTRAN:**

```
PARAMETER (K=1212)
DATA ITABLE /
  A 0, 2586, 0,
  B 5, 60000, 0,
  C 1, 574, 0,
  D 6, 9000, 0 /

BUFFER IN(10,1) (IARRAY(1),IARRAY(N) )
IF( UNIT(10) ) ....
-or-
CALL RDTAPE(10,1,0,IARRAY,N)
CALL IOWAIT(10,NSTATE,NWDS)

CALL MC0NV( IARRAY, ICARAY, K, ITABLE, 4, IER )
IF( IER .NE. 0 ) ...
```

**Notes** 1. In the example above, the record was converted in the following manner:

   a. The first 2586 bits (431 characters, or 44 CDC 7600 words) are converted from DPC to ASCII, inserting the ASCII characters into 53 Cray-1A words (431 chars/8 char per word).

   b. The next 60000 bits (1000 CDC 7600 words) are converted from CDC 7600 to Cray-1A integer, the resulting 1000 Cray-1A words are put into the Cray-1A array (CARRAY) beginning at word 54.

   c. The next 574 bits are taken from the CDC 7600 array and transferred bit for bit to the Cray-1A array beginning at word 1054.
d. The last 9000 bits (150 CDC 7600 words) are converted from CDC 7600 to CRAY-1A floating point values and are inserted beginning at word 1063 of the CRAY-1A array.

e. The length of the CRAY-1A array (K above) is therefore 1212 words.

2. For mode conversions 5 (integer), 6 (floating point), or 9 (60-bit to 64-bit right justified), the number of bits to be converted must be a multiple of 60 bits.

For mode conversions 0 (DPC) and 2 (BCD—not yet implemented), the number of bits to be converted must be a multiple of 6 bits.

For mode conversions 3 (ASCII) and 4 (EBCDIC—not yet implemented), the number of bits to be converted must be a multiple of 8 bits.

3. The input statement (BUFFER IN or RDTAPE) will actually attempt to read more words than are on the record; this will result in the entire record being read and does not constitute an error.

4. Note difference in statement structure of IF(UNIT) test between CDC 7600 and CRAY-1A.

5. If integer and floating point values are mixed and all integers are CDC 7600 48-bit integers, then mode 6 may be used to convert both.

6. The 556 and 200 bpi density cases run identically to the 800 bpi case above, with the following changes:

a. In the CDC 7600 JCL, the density on the *VOLUME or *ASSIGN command line would change to 5 for 556 bpi and 2 for 200 bpi in place of 8 for 800 bpi.

b. In the interactive session with TAPECY, the input tape mode would change to 6 for 556 bpi and 1 for 200 bpi, in place of 11 for 800 bpi.

7. If the tape had been written as a segmented tape (using unformatted FORTRAN write statements on the CDC 7600 as opposed to the BUFFER OUT/WRTAPE statements shown), it could be read by essentially the same job with the addition of the following lines to the CRAY-1A JCL:

Examples 6-28 October 1983
ACQUIRE, DN=DSEG, MF=TB.
DSEG, I=xoxoxo, O=yyyyy.
REWIND, DN=yyyyy.
ASSIGN, DN=yyyyy, A=FT10.

These lines should replace the ASSIGN statement in the example.

8. Currently, the maximum size of a record being read through the MF=MT, is less than or equal to 1,015,808 bytes.
Below is a list of some of the error messages users will receive. Those messages in the range MT001–MT099 are informative messages which may be issued many times during the course of processing a tape. No more than sixty-four such informative messages will be issued—any messages after the sixty-fourth will not be issued, although tape processing will continue.

If the user’s job is still in the system, the message will be placed in the job’s $LOG file. If the job has completed, the message will be written to the file ACCNT, and the ACCNT file will be disposed to the user, as is done by the TBM currently under similar conditions. If there are several informative messages, they will all be written to the same ACCNT file which will be sent to the user at the completion of processing. If several warning messages are being issued, COS tends to throw some of them away. Accordingly, once approximately six messages have been issued to the user’s logfile, all subsequent messages are sent to an ACCNT file even if the user’s job is still running.

The words shown in the messages as being in lower case will be replaced by their real values when the messages are sent to the user.

MT000  n FILES m RECORDS READ FROM vsn
       n FILES m RECORDS WRITTEN TO vsn
       n FILES m RECORDS READ FROM vsn WITH n RECOVERABLE ERRORS
       n FILES m RECORDS READ FROM vsn WITH u/r ERRORS
       n FILES m RECORDS READ FROM vsn WITH u UNRECOVERABLE ERRORS
       n FILES m RECORDS WRITTEN TO vsn WITH n RECOVERABLE ERRORS

This message indicates the successful completion of a staging request. It tells whether the tape was read or written and the number of errors, if there were any. If there were both recoverable and unrecoverable errors on an ACQUIRE request (note that there cannot be unrecoverable errors on a DISPOSE), the number of unrecoverable errors is printed first followed by the number of recoverable errors. The two will be separated by a slash.

MT001  READ n RECORDS OF m BYTES FROM vsn

The number of consecutive records of a given length read from the tape.

MT002  WROTE n RECORDS OF m BYTES TO vsn

The number of consecutive records of a given length written to the tape.

MT003  RECORD m FILE n ON vsn > blksize BYTES, RECORD TRUNCATED

The specified record number, which is part of the indicated file...
number, was longer that the specified (or default) BLKSIZE. The record is truncated. The vsn given is the name of the tape. This warning is only issued for the first truncated records. Subsequent truncations are not flagged, as the number of such warnings tends to be very large.

MT004 READ TAPE MARK FROM vsn AFTER RECORD n

Issued whenever a tape mark is read. Note that when reading a double tape mark which is being used to delimit the end of the data on the tape, this message is only issued once.

MT005 WROTE TAPE MARK TO vsn AS RECORD n

Issued whenever a tape mark is written.

MT006 USING BLKSIZE=n FOR vsn

Whenever the software automatically sets the BLKSIZE, this message is issued to tell the user what value has been used.

MT007 READ vsn WITH DEN=density

The user-specified (or defaulted) density on the tape that was just processed was incorrect. This warning message tells what the correct density specification was.

MT008 END-OF-TAPE DETECTED AFTER RECORD m FILE n ON vsn

The reflector at the end of the tape was detected. It is being ignored, since the user specified EOTP=YES.

MT101 OPERATOR CANNOT MOUNT tapeid

The operator has signalled that the requested tape cannot be mounted. The means one of the following:

1. You are attempting to write on someone else’s tape.
2. The specified tape name doesn’t exist.
3. Something else. Please contact an operator in the machine room for an explanation.

MT102 CANNOT SAVE pdn--ERROR CODE 000

Applies only to ACQUIRE requests. The software was unable to SAVE the dataset. This error message is very difficult to receive.

The only way known to receive this error is to specify both the ED and M parameters on the ACQUIRE request. ED must be such that while the specified edition doesn’t exist, other editions do. M must be
incorrect. The error code shown is the error code described in the documentation on the PDD status in COS EXEC/STP/CSP Internal Reference Manual, SM-0040.

MT103 CANNOT OPEN DISPOSED DATASET--ERROR CODE 000

The error code shown is the error code described in the documentation on the PDD status in COS EXEC/STP/CSP Internal Reference Manual, SM-0040.

MT104 RECORD m FILE n ON vsn < 12 CHARACTERS

Applies only to DISPOSE requests. It indicates that the specified record on the disk file was too short for the hardware to write. The vsn given is the name of the tape.

MT105 SYNTAX ERROR IN TEXT FIELD FOR vsn

There was something wrong with the parameters in the "TEXT=" field of the ACQUIRE or DISPOSE statement.

MT106 vsn REQUIRED KEYWORD key WAS MISSING
vsn OUTPUT KEYWORD OVERFLOW
vsn SYNTAX ERROR
vsn UNKNOWN OR DUPLICATE KEYWORD key
vsn UNEXPECTED SEPARATOR ENCOUNTERED
vsn key CANNOT BE EQUATED
vsn key MUST HAVE A VALUE
vsn MAXIMUM KEYWORDS EXCEEDED
vsn INVALID RETURN STATUS--CANNOT RECOVER

All these messages, which should be self-explanatory, come about as the result of the $SYSLIB routine PPL returning an error status. For more information, see the Library Reference Manual, SR-0014.

MT107 INVALID SPECIFICATION key=value FOR vsn

Self-explanatory. The most common causes for this message are the following:

1. A field which should have been numeric wasn't (e.g., ERRMAX=0 instead of ERRMAX=0).

2. A number was too large (e.g., BLKSIZE > 1015808, which is 126,976, 64-bit words).

3. A keyword which can have only certain values, such as "YES" or "NO", has some other value.
MT108 LRECL=lrecl > BLKSIZE=blksize FOR vsn

Issued if LRECL is greater than BLKSIZE.

**NOTE:** This error message (and the LRECL parameter) has not yet been implemented.

MT109 UNKNOWN FORMAT DF=df FOR vsn

Issued if the value specified for DF on the ACQUIRE or DISPOSE statement is not one of the allowable values. The allowable values are described in Chapter 3 of this guide.

MT110 TAPE I/O ERROR ON vsn RECORD n FILE m. STATUS=n

Issued if a fatal tape I/O error is encountered.

MT111 END-OF-TAPE DETECTED AFTER RECORD m FILE n ON vsn

The reflector at the end of the tape was detected.

MT112 DISK I/O ERROR ON pdn

An unrecoverable I/O error occurred while reading vsn from the disk or writing it to the disk.

MT113 EOD ENCOUNTERED WHILE PROCESSING SKIPF/SKIPR FOR pdn

While skipping the requested files and/or records at the beginning of a DISPOSEd dataset, the end of the dataset was encountered.

MT114 RECORD m ON vsn IS m BITS LONG WITH DF=TR

There are two possible reasons for this message. Either m is not a multiple of 32,768 bits (512 words), or the tape contains variable length records. All the records except the last one on a transparent mode tape must be the same length. The last records may be shorter than the rest, but it still must be a multiple of 512 64-bit words.

MT115 BLOCK NUMBER ERROR ON vsn

While reading vsn to process a DISPOSE request, a Block Number Error (SL004) occurred.

MT116 DISK READ ERROR n ON vsn

While reading vsn to process a DISPOSE request, an error occurred. The error message number is n. For example, if n is 6, the error is SL006, which is a prematurely terminated dataset.

MT117 DATA CHECK ON vsn AT RECORD n FILE m

While reading the indicated tape, there was a non-recoverable "data
check" error. This is usually a parity error and indicates a bad spot
or improperly written record on the tape.

If the analagous situation occurs while writing a tape, the user is
not informed—the operator is told to re-start the operation with a
different tape drive and/or a different reel of tape.

MT118 INTERFACE CONTROL CHECK ON vsn AT RECORD n FILE m

If this error occurs while reading a tape, it is not treated as a
fatal error. This situation has been found in practice to almost
always mean that there just isn’t any more data to be read on the
tape. In this case, the reading of the tape is stopped, and the user
ACQUIRE request is signalled complete with as much data as the program
was able to read. Note that if the last thing read from the tape was
a data record rather than a file mark, the COS disk I/O routines
automatically write an end-of-file to the disk when the disk dataset
is closed. This message is usually followed by the MT001 message tel-
ling how many records were read before the interface control check
occurred.

At this time, results are uncertain if this error occurs while writing
a tape.

MT119 BAD RECORD ON vsn AT RECORD n FILE m

If this error occurs while reading a tape, it is not treated as a
fatal error. The erroneous record will be flagged as such when it is
written to the disk dataset being ACQUIREd.

MT120 TAPE OFF REEL ON vsn AT RECORD n FILE m

If this error occurs while reading a tape, it is not treated as a
fatal error. The data on the tape has been processed correctly up to
this point, there just wasn’t enough tape left after the last record
read to fill the vacuum columns. The reading of the tape is stopped,
and the user ACQUIRE request is signalled complete with as much data
as the software was able to read. Note that if the last thing read
from the tape was a data record rather than a file mark, the COS disk
I/O routines automatically write an end-of-file to the disk when the
disk dataset is closed. This message is usually followed by the MT001
message telling how many records were read before the tape ran off
the end of the reel.

MT121 NO DATA ON TAPE vsn

This is indeed a fatal error, and very difficult to receive.
The following is an example of the logfile output that might occur as the result of writing a tape:

11:09:44 8.0684 CSP DISPOSE(DN=B12345,DF=BI,MF=MT,TEXT="DEN=1600",WAIT)
11:09:56 8.0725 TBT MT002 - WROTE 10 RECORDS OF 54 BYTES TO B12345
11:09:56 8.0789 TBT MT002 - WROTE 15 RECORDS OF 47 BYTES TO B12345
11:09:56 8.0845 TBT MT005 - WROTE TAPE MARK TO B12345 AS RECORD 26
11:09:56 8.0850 TBT MT005 - WROTE TAPE MARK TO B12345 AS RECORD 27
11:09:56 8.0856 TBT MT000 - 2 FILES 27 RECORDS WRITTEN TO B12345
APPENDIX B

TEIM Error Messages

The following is a list containing all messages that can be issued by the Ter-
abit Memory System (TBM). Many of these will be seen only by the operators or
systems programmers. Any error that a user can receive is followed by a
description of the probable cause and a recommended course of action.

TB001 COULD NOT LOCATE VSN ON RETURN FROM TBM
Possible cause: CRAY-1A job terminated before TBM job returned.
Action: Re-submit.

TB002 INCORRECT READ PASSWORD, CANNOT ACCESS 'VOLUME'
Possible cause: ACQUIRE specified different read password from pass-
word in directory. Action: Find out the correct password and re-
submit.

TB003 WRITE
Cause: Explanatory message only; no problems.

TB004 RETURN OF UNKNOWN OP FROM DISPLAY STATUS

TB005 READ
Cause: Explanatory message only; no problems.

TB006 READ PASSWORD ERROR IN DISPOSE
Possible cause: DISPOSE specified different password from that in
directory. Action: Determine correct password and re-submit.

TB007 WRITE PASSWORD ERROR IN DISPOSE
Possible cause: DISPOSE specified different password from that in
directory. Action: Determine correct password and re-submit.

TB008 SPECIFIC WRITE NOT ALLOWED

TB009 TBM VSN NOT IN DIRECTORY
Possible cause: Incorrect VSN name specified, or problem with TBM
directory. Action: Determine correct VSN name and re-submit. If name
is correct, check fiche listings of all known TBM volumes in Consult-
ing Office library. If it is listed yet unavailable, see the TBM spe-
cialist.

TB010 INSUFFICIENT SPACE FOR DIRECTED DESCEND
Possible cause: MVN does not have enough room remaining to write this
new dataset. Action: Clean up the tape or get another one.

TB011 VOLUME DOES NOT EXIST

TB014 DIRECTORY FOR VOLUME TABLE OF CONTENTS 'NOT IMPLEMENTED'
TB015 INCORRECT DIRECTORY REQUEST

TB016 TBM DATA SPACE OVERFLOW - RETRY ATTEMPTED
Action: If you receive this message, contact Dave Kitts.

TB017 SCP FAILURE IN TBM - RETRY ATTEMPTED
Action: If you receive this message, contact Dave Kitts.

TB018 CIU FAILURE IN TBM - RETRY ATTEMPTED
Action: If you receive this message, contact Dave Kitts.

TB019 DATA CHANNEL ERROR IN TBM - RETRY ATTEMPTED
Action: If you receive this message, contact Dave Kitts.

TB020 TRANSPORT DRIVER FAILURE IN TBM
Possible cause: TBM hardware problem. Action: check machine status with consultants, then re-submit.

TB021 TRANSPORT FAILURE IN TBM
Possible cause: TBM hardware problem. Action: check machine status with consultants, then re-submit.

TB022 DATA ERROR IN TBM
Possible cause: TBM hardware problem. Action: check machine status with consultants, then re-submit.

TB023 FATAL ERROR - JOB TERMINATED
Action: If you receive this message, contact Dave Kitts.

TB024 NOT IN MSD DIRECTORY
Possible cause: Incorrect VSN name specified, or problem with TBM directory. Action: Determine correct VSN name and re-submit. If name is correct, check fiche listings of all known TBM volumes in Consulting Office library. If it is listed yet unavailable, see the TBM specialist.

TB025 TBM INITIAL 45 SECOND DELAY

TB026 ABORT / MASTER CLEAR ISSUED FOR TBM

TB027 ACQUIRE CANNOT BE USED FOR DELETE
Possible cause: Tried to specify TEXT=“DELETE” on ACQUIRE. Action: Use DISPOSE instead of ACQUIRE.

TB028 DELETED VOLUME
Cause: Explanatory message only; no problems.

TB029 CHECKSUM ERROR FOUND WHILE VERIFYING VOLUME

TB030 IBM TIME OUT, SET SSW 3 TO CONTINUE

TB031 PROCESSING RESUMED, SSW 3 IS OFF

TBM Error Messages

Appendix B-2

October 1983
TB032 SSW2 SET, TO RESUME PROCESSING CLEAR IT

TB033 SSW2 CLEARED, PROCESSING REINITIALIZED

TB034 DISPOSE COMPLETED FOR VOLUME

Cause: Explanatory message only; no problems.

TB035 TEM BLOCKS WRITTEN TO

Cause: Explanatory message only; no problems.

TB036 TEM BLOCKS READ FROM

Cause: Explanatory message only; no problems.

TB037 WRITE PASSWORD INVALID, WRITE SEEK WAS ISSUED

TB038 READ PASSWORD INVALID, READ SEEK WAS ISSUED

TB039 INVALID PARAMETERS. THE PARAMETERS SENT BY THE CRAY-1A ARE INVALID.

TB040 THE CRAY-1A REQUESTED THE TRANSMISSION OF MORE THAN 43800 TEM BLOCKS

Possible cause: User requested more data than can fit on all the CRAY-1A disks. Action: If you don’t know how to free up some space, see a consultant.

TB041 THE PARAMETER LENGTH IS NOT COMPATIBLE WITH THE COMMAND

TB042 AN UNRECOGNIZED COMMAND CODE WAS RECEIVED BY TEM FROM THE CRAY-1A

TB043 THE CRAY-1A HAS NOT TRANSMITTED THE NUMBER OF PARAMETERS SPECIFIED

TB044 THE CRAY-1A TRANSMITTED MORE BLOCKS THAN WERE SPECIFIED

TB045 OFF-LINE, COPY PUT TO THE PRODUCTION TAPE

Cause: Explanatory message only; no problems.

TB046 WAS IN CDC 7600 FORMAT WHEN RECEIVED FROM THE TEM.

Cause: Explanatory message only; no problems.

TB047 TEM WRITE FAILED. POSSIBLE CIU FAILURE.

TB048 TEXT FIELD HAS INCORRECT FIELD DELIMITERS

Possible cause: CRAY-1A did not receive single quote as TEXT field delimiter. Action: Check TEXT fields on ACQUIRE and/or DISPOSE cards for invalid or extraneous characters.

TB049 INCORRECT PARAMETER IN TEXT FIELD

Possible cause: A parameter was specified in the TEXT field which is not a recognized keyword. Action: Check documentation or see a consultant for list of recognized keywords.

TB050 MVN PARAMETER HAS MORE THAN 6 CHARACTERS

Possible cause: Dedicated tape name is longer than 6 characters. Action: Use the correct tape identifier.
TB051  CRPASS PARAMETER HAS MORE THAN 8 CHARACTERS
Possible cause: Read password contains too many characters. Action: Use a password of 8 or fewer characters.

TB052  CWPASS PARAMETER HAS MORE THAN 8 CHARACTERS
Possible cause: Write password contains too many characters. Action: Use a password of 8 or fewer characters.

TB053  DIR PARAMETER HAS MORE THAN 6 CHARACTERS
Possible cause: Dataset name specified to receive directory from TBMOONV contains more than 6 characters. Action: Use a dataset name of 6 or fewer characters.

TB054  KEY PARAMETER HAS MORE THAN 8 CHARACTERS

TB055  NO CONVERSION REQUESTED
Cause: Explanatory message only; no problems.

TB056  AUTOMATIC CONVERSION REQUESTED
Cause: Explanatory message only; no problems.

TB057  SDT ENTRY NOT FOUND, REINITIALIZE

TB058  CHANNEL ERROR, SET SSW3 TO CONTINUE

TB059  ABORT ISSUED.

TB060  UNABLE TO OPEN SPOOLED DATASET, THE NAME MAY BE SAME AS JOB NAME
Possible cause: Job name is the same as a dataset name. Action: Change job name and re-submit.

TB061  MASTER CLEAR ISSUED

TB062  INVALID CHARACTER IN VSN NAME
Possible cause: Invalid or extraneous character in dataset name. Action: Check dataset names in DISPOSE/ACQUIRE statements to insure that they contain 8 or fewer characters, that the first character is alpha-numeric or $,@, or %, and that the remaining characters are alpha-numeric. All alphabetic characters should be uppercase.

TB063  TOO MANY CHARACTERS IN VSN NAME
Possible cause: Extra character(s) in dataset name. Action: Check dataset names in DISPOSE/ACQUIRE statements to insure that they contain 8 or fewer characters.

TB064  TIME OUT DURING TEM BUFFER WRITE ON VSN

TB065  TIME OUT DURING TEM BUFFER READ ON VSN

TB066  SENSE SWITCH 4 SET, PROCESSING DIRECTORY CALLS ONLY

TB067  WRITE PASSWORD ERROR UPDATING VSN
Possible cause: Write password missing or doesn’t match the password
in the directory for the dataset specified on a DISPOSE with
TEXT="UPDATE". Action: Determine correct password and re-submit.

TB068  UPDATED VSN
Cause: Explanatory message only; no problems.

TB069  INCORRECT SPECIFIC READ REQUEST

TB070  MVN NOT SPECIFIED FOR SPECIFIC READ

TB071  SCIENTIST NUMBER NOT ALL DIGITS
Possible cause: One of the first four characters following 'US=' on
the JOB statement is non-numeric. Action: Correct US=xxxxppppppppp on
your JOB statement and resubmit.

TB072  INPUT CHANNEL ADDRESS ERROR

TB073  OUTPUT CHANNEL ADDRESS ERROR

TB074  OUTPUT CHANNEL ERROR

TB075  INPUT CHANNEL ERROR

TB076  READY FLAG NOT EXPECTED

TB077  DONE FLAG NOT EXPECTED

TB078  BUSY FLAG NOT EXPECTED

TB079  INTERFACE STATUS MALFUNCTION

TB080  DONE FLAG EXPECTED

TB081  UNKNOWN ERROR IN INTERFACE STATUS

TB082  TBM STATUS IS NOT READY AND NOT BUSY

TB083  MALFUNCTION IN INTERFACE STATUS

TB084  UNABLE TO WRITE MESSAGE TO JXT

TB085  MVN NOT IN DIRECTORY
Possible cause: Incorrect MVN name specified, or problem with MVN
directory. Action: Determine correct MVN name and re-submit. If name
is correct, see the TBM specialist.

TB086  UNRECOVERABLE DATA ERROR READING CRAY-1A DISK FOR VOLUME
Action: If you receive this message, contact Dave Kitts or Gene
Schumacher.

TB087  UNRECOVERABLE HARDWARE ERROR READING CRAY-1A DISK FOR VOLUME
Action: If you receive this message, contact Dave Kitts or Gene
Schumacher.
TB088  TEM INDICATES PERMANENT ERROR IN VOLUME
Possible cause: Both the working copy and the backup copy of your
dataset are bad. Action: See the TEM specialist.

TB089  TMS4 PROCESSING DIRECTORY READ"*L

TB090  TMS4 PROCESSING DIRECTORY WRITE

TB092  START READ OF

TB093  START WRITE OF

TB094  GENERIC RESOURCE NOT DECLARED FOR DIRECTED DISPOSE OF VSN
Possible cause: JOB statement does not contain *TD parameter. Action:
Add a *TD parameter to your JOB statement and resubmit.

TB095  DATASET IS ALREADY ON THE DISK
Cause: Explanatory message only; no problems.
APPENDIX C
Using the Data Management Support Processor (DMSP)

INTRODUCTION

The Data Management Support Processor (DMSP) is a Cray Research, Inc. software routine which permits 1/2-inch tapes to be ACQUIRED by the CRAY-1A computers from the tape drives connected to the IBM 4341 and/or DISPOSED from the CRAY-1A computers to the IBM 4341 tape drives. The software is not supported by Cray Research, Inc., or NCAR. DMSP is reserved primarily for use by the Systems Group within SCD. Its use is discouraged for general users.

CRAY-1A access to tape data will be through the IBM 4341 system. If you wish to use tape data on the CRAY-1A, you must first move the data to the CRAY-1A disks. If you wish to create tape data from CRAY-1A programs, you must create disk files and then move data to the tape. In most cases, this can be done with normal ACQUIRE and DISPOSE statements. Tapes to be read (written) in this way must (will) meet the following specifications:

1. 9-track.

2. Terminated with two consecutive file marks.

To handle tapes which do not meet these specifications, it will be necessary to do a preprocessing step on the IBM 4341. This may consist either of creating a tape which does meet these specifications and acquiring the new tape, or reading the tape to the IBM 4341 disks and shipping the disk file to the CRAY-1A via the local network. The tape-to-tape option will normally be much simpler and should be used except where large numbers of tapes (more than 10) are involved. The local network option will require sending the volume to the TBM to insure that it is not lost before the CRAY-1A job attempts to access it. Volumes shipped across the local network must be smaller than 50 MB so that it will be necessary to segment full 6250 bpi tapes. These options are discussed in more detail in the following paragraphs. Note that we are only discussing moving physical tape blocks. Other compatibility problems such as data blocking, internal character set or work format differences, etc., must be addressed separately, usually through subroutines which perform the needed conversions.
Most situations where tapes cannot be acquired from the CRAY-1A can be solved by creating a new tape using TAPECY on the IBM 4341. Tapes with parity errors can be copied to remove errors, 7-track tapes can be copied to 9-track, and tapes without double file marks can be copied and double file marks inserted. In some cases it may be necessary to write a small FORTRAN program to create the compatible tape. To avoid compatibility problems with IBM 4341 tape formats and for faster execution, it is recommended that the RDTAPE/WRTAPE routines be used rather than IBM FORTRAN I/O statements.

For jobs with large numbers of tapes, the logistics problems of recopying all tapes to compatible tapes may be significant. It is possible to move tape data to the IBM 4341 disks and then ship the disk file across the local network. One option is to use CMS formatted disks, move the tape data to a named CMS file, and ship the file across the network. Due to very slow disk performance, this option is only suitable for small character files such as programs or formatted data files, and these are more easily handled using ACQUIRE statements. When this option is used, it may be implemented using standard CMS commands and the NET command.

The only practical way that files of the size of a 1600 bpi tape can be sent across the local network is by using special fast disk routines which allow writing on an unformatted disk, directly in network format. These routines are still in the development stages and users needing this capability should contact the SCF consultants.

In order to retrieve a dataset from mag tape, the CRAY-1A user must use an ACQUIRE request. The following parameters are required or supported:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DN=dn</td>
<td>The local dataset name (dn) is the name the dataset will be known by once the ACQUIRE has been completed.</td>
</tr>
<tr>
<td>PDN=pdn</td>
<td>The permanent dataset name (pdn) is the name the dataset will be known by on the CRAY-1A. If this parameter is not specified, the local dataset name (dn) will be used. This name must correspond to the 1/2-inch tape labels.</td>
</tr>
</tbody>
</table>

Note: The ACQUIRE function first checks the CRAY-1A Dataset Catalog. If the specified dataset already exists on the CRAY-1A, the
ACQUIRE is converted into an ACCESS and the resident CRAY-1A dataset is used. If the requested dataset does not exist on the CRAY-1A, a request for the dataset is sent to the specified mainframe. When the dataset is returned to the CRAY-1A, it is saved (i.e., made a permanent dataset). The ACQUIRE is then turned into an ACCESS and that dataset is accessed.

**MF=I2** The mainframe identifier I2 is required to indicate that the dataset is to be acquired from the IBM 4341.

**UQ** The unique access parameter (UQ) must be specified if the user is going to write on the dataset or use the MODIFY or DELETE services on the permanent dataset.

**TEXT=** The TEXT parameters provide descriptive information about the tape. These parameters are discussed below.

**DF=df** The DF parameter may be either TR, BB, or CB. If TR is specified, the tape is read as a continuous byte string, with consecutive tape blocks concatenated together on byte boundaries. Tapes ACQUIRED in this manner will usually be those DISPOSED with a DF=TR. If BB or CB is specified, the COS blocked structure is created, with each tape block becoming a COS record, and tape marks becoming EOF RCWs (Record Control Words). A null file (tape mark with no proceeding data) becomes an EOD RCW. If CB is specified, the data is translated from EBCDIC to ASCII. The default is CB.

**DISPOSITION TO MAGNETIC TAPE** In order to write a dataset on mag tape, the CRAY-1A user must use a DISPOSE request. The following parameters are required or supported:

<table>
<thead>
<tr>
<th>DISPOSE, DN=dn, SDN=sdn, DC=MT, MF=I2, DF=df, WAIT, DFER, NRLS, TEXT=&quot;text parameters&quot;</th>
</tr>
</thead>
</table>

**DN=dn** The dataset name (dn) is the local dataset name of the dataset to be disposed. This is a required parameter.

**SDN=sdn** The staged dataset name (sdn) is the tape name which will appear in the tape mount request message on the IBM 4341 operator console. If
this parameter is omitted, the local dataset name (dn) will be used.

DC=MT
The disposition code MT is required to indicate that the dataset is to be written to mag tape. (The default disposition code is PR, which specifies that the dataset is a print file.)

MF=I2
The mainframe identifier I2 is required to indicated to COS that the dataset is to be disposed to the IBM 4341.

DF=df
The dataset format may be specified as either TR, BB, or CB. The TR format instructs the Data Management Support Processor (DMSP) to write the data as a string of bytes without looking at the data. (The data is written on tape in fixed-length blocks equal in length to the disk segment size.) The BB and CB formats signify that the data is in COS blocked format, either binary or coded ASCII. A tape block is written for each COS record. EOF and EOD record control words are recorded as tape marks. For CB format, the data is translated to EBCDIC before writing it to tape.

Any DF value other than BB, CB, or TR will cause the job to abort. Remember DF=CB is the default, but most of the time you will use DF=BB.

WAIT
If the WAIT parameter is specified, the user program will not resume execution until the dataset has been written successfully on mag tape.

DEFER
The DEFER option indicates that the DISPOSE is to be deferred until the dataset is released. This parameter is used most often to postpone disposition of the dataset until the job terminates or to guarantee disposition of the dataset if the job aborts. The default is to dispose the dataset at the time the DISPOSE request is encountered.

NRLS
The no release option (NRLS) indicates that the dataset is not to be released after it is disposed. The dataset then remains available for additional processing by the job after the tape is written. The default is to release the disk space occupied by the dataset once the DISPOSE has been completed.
The TEXT parameters provide descriptive information about the tape. These parameters are discussed below.

The format of the text field on ACQUIRE and DISPOSE statements when using magnetic tapes is:

```
TEXT='BLKSIZE=blks,DEN=den,NFILES=nf,ERRMAX=emx,RTRYCNT=rtr'
```

**IMPORTANT NOTE:** The following "TEXT=" keywords and their values are the only parameters recognized as valid in the "TEXT=" field of an ACQUIRE or DISPOSE statement by DMISP.

- **BLKSIZE=blks** A decimal integer greater than 12 and less than 520192 which is the maximum size of physical tape records in bytes. Default size is 32768.

- **DEN=den** Tape density in bits-per-inch. 800, 1600 and 6250 are supported. The default is 6250.

- **NFILES=nf** Maximum number of files to be read from tape. If NFILES is not specified the tape is read until two consecutive file marks are reached. NFILES is ignored if DF=TR.

- **ERRMAX=emx** Maximum number of unrecoverable read errors permitted before the request is aborted. The default is 0. Applies to ACQUIRE only.

- **RTRYCNT=rtr** Number of unsuccessful attempts to read a record before the record is flagged as being in error. Default is 0 which indicates no error processing is done. Applies to ACQUIRE only.

An example of one possible format for the "TEXT=" field is as follows:

```
TEXT='BLKSIZE=4096,DEN=1600,NFILES=12'
```

**EXAMPLES:** MOVING DATA BETWEEN THE IBM 4341 TAPE DRIVES AND THE CRAY-1A COMPUTERS

In this section we will provide some examples which demonstrate how to move datasets between the IBM 4341 tapes and CRAY-1A disks. Reading and writing of tapes on
the IBM 4341 from the CRAY-1A is done through the fast link, MF=I2. Again, the ellipses ( ..... ) represent the job step(s) which accessed/created the CRAY-1A dataset.

**CRAY-1A Disk to IBM 4341 Tape**

**Example 1:** A job to stage a CRAY-1A dataset to 9-track, 1600 bpi, 1/2-inch tape on the IBM 4341 with no conversion of the CRAY-1A dataset. CRAY-1A block structure is removed, and logical records written can be no more than 32K-bytes. **NOTE:** DC=MT must be specified or the dataset will go to the printer instead.

```
JOB,JN=ANAME,US=111122222222,*I2.
          *************
DISPOSE,DN=B01234,DF=BB,TEXT='DEN=1600'.
    \EOP
```

**Example 2:** A job to DISPOSE a CRAY-1A disk dataset to a 9-track, 6250 bpi 1/2-inch tape on the IBM 4341 in CRAY-1A format. **Note:** DC=MT is neccessary to go to tape instead of printer.

```
JOB,JN=ANAME,US=111122222222,*I2.
          *************
DISPOSE,DN=MYJOB1,SDN=B01234,MF=I2,DC=MT,DF=TR,TEXT='DEN=6250'.
    \EOP
```

**Example 3:** A job to make a 9-track 800 bpi tape of card images of a program on a CRAY-1A disk dataset named MYJOB1. The dataset must have been created with BFI=OFF in order to remove the blank compresssion on the CRAY-1A dataset. The dataset will be written as unblocked card images. There will be a conversion of ASCII to EBCDIC. **Note:** DC=MT must be specified to write tape instead of the printer.
IBM 4341 Tape to 
CRAY-1A Disk

Example 4: A job to read a 1600 bpi 9-track tape on the IBM 4341 and stage it to CRAY-1A disk with no conversion of information. Conversion to CRAY-1A block structure will occur. The physical records may be as large as the IBM 4341 default of 32768 bytes. Data ends on standard double end-of-file (EOF). The job will abort on any unrecoverable read errors.

Example 5: A job to read a 6250 bpi 9-track tape on the IBM 4341 and stage it to the CRAY-1A disk. The records are in CRAY-1A format so no blocking is necessary. The IBM 4341 block-size is set to the maximum allowed.

Example 6: A job to read a "dirty" 1600 bpi, 9-track tape containing EBCDIC characters. Records are under the default block size, conversion to ASCII is requested. The software will attempt to reread a "bad" record 20 times, will allow 1000 unrecoverable errors, and will read only two files.
Example 7: IBM 4341 9-track Tape to CRAY-1A Disk to CRAY-1A Tape

```
JOB,JN=ANAME,US=111122222222,*I2,*MT.
ACQUIRE, DN=B07600,MF=I2,DF=BB,TEXT="DEN=1600".
DISPOSE, DN=B07600,MF=MT,DF=BB,SDN=B07700.
\EOF
```

Example 8: CRAY-1A 1/2-inch Tape to CRAY-1A Disks to IBM 4341 Tape

```
JOB,JN=ANAME,US=111122222222,*I2, *MT.
ACQUIRE, DN=B07600,MF=MT,DF=BB.
DISPOSE, DN=B07600,MF=I2,DC=MT,DF=BB,TEXT="DEN=1600".
\EOF
```

Example 9: IBM 4341 9-track Tape to CRAY-1A Disk to the TBM

```
JOB,JN=ANAME,US=111122222222,*I2,*TB.
ACQUIRE, DN=B07600,MF=I2,DF=BB,TEXT="DEN=1600".
DISPOSE, DN=B07600,MF=TB,DF=BB.
\EOF
```

Example 10: TBM to CRAY-1A Disk to IBM 4341 9-track Tape

```
JOB,JN=ANAME,US=111122222222,*I2,*TB.
ACQUIRE, DN=B07600,MF=TB,DF=BB.
DISPOSE, DN=B07600,MF=I2,DF=BB,DC=MT,TEXT="DEN=1600".
\EOF
```
GLOSSARY

Ascending
This term is used to describe dataset movement to the CRAY-1A computers from the Terabit Memory System (TBM) or the IBM 4341.

Binary bit serial
All data is written in this manner. The term is used to describe a method of data transmission in which only the data bits are present (i.e. no blocking information).

Bit
Binary digit. The digits are always represented by alternate states (e.g. 1 or 0, ON or OFF, + or −) and are used to carry and process information in modern digital computing machines.

Blocking
Blocking, physical records, and logical records are three interrelated terms which need to be understood together. On a data storage device there is usually an optimum record size which determines the size of the physical records for that device. On the CRAY-1A disks, this record size is 4096 bytes and is called a disk sector. However, different devices have different optimum record sizes and programmers may not always use records of the same length even if there is a standard. The record that the user writes is called the logical record. The operating system must find a way to fit the user’s logical records into the device specific physical records, and this process is referred to as blocking. The operating system adds additional information such as record counters and/or pointers to the next physical record to each physical record it creates from your logical records. This is known as blocking information and is generally transparent to the user.

BPI
Bits Per Inch. The number of binary digits which can be placed on an inch of 1/2-inch wide magnetic tape. The bpi figure is used to indicate the density of the tape for recording information (e.g. 6250 bpi tape means that the tape can hold 6,250 bits per inch).

Byte
A byte is 8 bits on some computers. A byte may be used to contain the code for a single character, such as the letter ‘A’, the number ‘7’, or a special symbol such as a square bracket ‘[‘.

Character
A character may be understood on several levels. A character is a symbol, a way of transferring information. There are alphabetic characters, numeric characters, special characters (+,-,/) and control characters. Since...
computers only deal in bits, each character is translated into some sequence of 1's and 0's according to certain translation tables. There are several different types of translations, among them ASCII (used by the CRAY-1A mainframes and the NCAR Local Network), EBCDIC (used by the IBM 4341 machines), and External BCD which was used by 6 bit-word physical devices (card readers, magnetic tape drives, the now de-commissioned CDC 7600, etc.).

Checksum

A checksum is a way for the operating system to test for data transmission errors. It is a logical sum of bit segments of transmitted data. (Logical sum means they either add up to 1 or 0, depending if the algebraic sum is odd or even and according to local operating system conventions).

Control point

Control point is a term used for computers with multiprocesssing operating systems. It describes an actual execution slot for a job, and keeps track of the status of that job as it is running.

Dataset

A dataset is a collection of records containing information. The records of a dataset may be partitioned into a series of files which may be either labeled or unlabeled. For our purposes, all datasets at NCAR are unlabeled CRAY-1A datasets.

Descending

This term is used to describe dataset movement from the CRAY-1A computers to the TBM or the IBM 4341.

Disk sector

A data storage disk is divided into wedges, known as disk sectors. On the CRAY-1A disks, a disk sector contains 4096 bytes. No smaller segment than a sector can be addressed by the disk drives in the direction of rotation of the disk. The smallest addressable segment in the radial direction of the disk is a single track.

Entry point

The place in memory at which a program code begins execution, or to which execution is transferred. At the user level, for example, a call to a subroutine can be an entry point to a particular code.

File

A file is a collection of records.

JCL

An acronym for Job Control Language. It refers to Cray Operating System commands (such as ACQUIRE, ASSIGN, CFT, LDR) which direct the execution of a job on one of the CRAY-1A computers.

Local Datasets

This term refers to datasets which exist ONLY for the duration of the execution of your computing job on one of the CRAY-1A machines. "Local" refers to the fact that the dataset is on the CRAY-1A disks and can be accessed by your job.
Logical record This is the series of bits, written by a single WRITE statement in a user's program. See also the definition of blocking.

Operating System A program on a computer which runs while the machine is operating. The Operating System manages details such as memory allocation, granting execution slots, handling I/O requests from executing programs, and numerous other details.

Permanent Datasets This term refers to datasets which are placed on CRAY-1A disks for use by your job during execution. They remain on the CRAY-1A disks for a pre-determined period of time (generally about 6 hours or less) depending upon disk loads on the CRAY-1A computer systems.

Physical record This is a machine dependent, fixed-size data segment of bits. The user's logical records are blocked into physical records by the operating system on the specific data storage device. See also the definition of blocking.

Record A record is a collection of bits, bytes, or words. For our purposes, all records are logical records (which may exist in a variety of formats) and are treated by the software of the Mass Storage Subsystem differently, depending upon the physical device on which the record will be mapped.

Segmented Segmentation is similar to blocking in that it is a mapping of the user's records into different size records. Segmentation refers specifically to tapes and was devised for efficient error recovery and minimum loss of data at a time when tapes were not exceedingly reliable. Segmented tapes tend to be divided into fairly small data segments.

Spooling Spooling is primarily an IBM invention created to allow reader and printer files to be loaded on/off a disk while other tasks are being executed on the computer. Dataset transfers to the IBM 4341 from the CRAY-1A computers are made to your "reader spool". You can then load them from the "reader spool" disk area to your IBM 4341 disk.

Staging Staging refers to the movement of datasets between computers and storage devices located on the NCAR Local Network. "Staging a dataset to the TBM" means that you are moving a dataset from one device on the Network to the Terabit Memory System. Staging is accomplished by Job Control Language (JCL) or user-specific calls to system software.
**TE4 block**  This is another NCAR-defined constant, equal to 30 CRAY-1A disk sectors, or 122880 bytes, or 15360 CRAY-1A (64-bit words) or 16384 CDC 7600 (60-bit words).

**Terabit Memory System (TBM) Data Envelope**  A TBM-specific process by which your dataset is partitioned into segments useful to the speed and read/write efficiency of the TBM. The envelope is a 1.2 megabit segment consisting of a header label block, a block of your data, and an end label block.

**Unblocked**  Term used to define a very special dataset whose records have lengths in integral multiples of 4096 bytes and which contain no blocking information. To access records in this type of dataset requires an assembly language routine which keeps track of sector/track addresses of each record in the disk dataset.

**Volume**  A volume is the same as a dataset. Datasets created on the CDC 7600 mainframe (now decommissioned) were referred to as volumes. Since this machine no longer exists at NCAR, we use the preferred term "dataset" which is appropriate to the CRAY-1A mainframes.

**VSN**  Acronym standing for Volume Serial Name. An identifying number (some people also use this term for named datasets) for your TBM volumes or datasets.

**Word**  This is a grouping of bits, and is machine dependent. It is the number of bits contained in one memory location. The CRAY-1A machines use 64-bit words, the CDC 7600 used 60-bit words, and the IBM 4341 uses 32-bit words.
INDEX

1/2-inch Tape, 2-8
CRAY-1A, 1-4
IBM 4341, 1-4
7600 Data,
Conversion of CDC, 4-25
7600 Datasets,
CDC, 3-7
7600 TBM Volumes,
Routines for Converting, 4-19
Access on MSS Devices,
File and Record, 2-8

ACQUIRE
Statement, 3-2
a CRAY-1A Dataset from TBM, 6-3
a Dataset from the IBM 4341, 6-5
and DISPOSE JCL Statements, 2-8
and DISPOSE Parameters, 3-6
TBM Dataset Written
on CDC 7600, 6-4

AMPEX Terabit Memory System (TBM), 2-1
Archival Dedicated Reel, 5-5
Archiving Datasets on the TBM, 5-3
Assistance,
Further, 5-11
Binary CMS Files on the TBM, 3-19
Blocked Record Format, 2-5
BRANIO: Direct (Random) I/O, 4-3

C CDC 7600 Data,
Routines for the Conversion, 4-25

C CDC 7600 Datasets, 3-7
CMS Files on the TBM,
Binary, 3-19
CMS Text and Program Files, 3-17
Command Syntax Review, 6-1

Computers,
Twin CRAY-1A, 1-4
Staging Jobs on the CRAY-1A, 3-1
Connection,
Cray Direct Tape, 3-8
Conventions,
TBM Dataset, 2-4
Conversion of CDC 7600 Data,
Routines for the, 4-25
Conversion Routines,
Data, 4-19
Conversion Routines,
Input/Output and Dataset, 4-1
Converting 7600 TBM Volumes,
Routines for, 4-19
Converting 7-track Tape to 9-track
Tape on the IBM, 6-23

COS-Supported Keywords, 3-14

CRAY-1A 1/2-inch Tapes, 1-4
CRAY-1A Dataset from TBM,
ACQUIRE a, 6-3
CRAY-1A Dataset to the TBM,
DISPOSE a, 6-3
CRAY-1A Dataset to the IBM 4341,
DISPOSE a, 6-5
Cray Direct Tape,
Examples Using, 3-16
Cray Direct Tape Connection, 3-8
CRAY-1A Disk,
Reading a CRAY-1A Tape to a, 6-6
CRAY-1A Tape to a CRAY-1A Disk,
Reading a, 6-6
CRAY-1A Computers,
Staging Jobs on the, 3-1
CRAY-1A Dataset Names, 2-3
CRAY-1A Datasets, 2-3, 3-6
CRAY-1A Disk, 2-8
CRAY-1A JCL Used to
Move Datasets, 3-2
Data Conversion Routines, 4-19
Data from Tape on Fiche,
Print Character, 3-22
Dataset,
Disposition of, 3-3
Dataset and File Names,
Special Problems with TBM, 2-5
Dataset Conventions,
TBM, 2-4
Dataset Conversion Routines,
Input/Output and, 4-1
Dataset Definitions, 2-2
Dataset Format, 3-3
Dataset Formats,
Other, 2-7
Dataset Formats,
Segmented, 2-7
Dataset from TBM,
ACQUIRE a CRAY-1A, 6-3
Dataset from the IBM 4341,
ACQUIRE a, 6-5
Dataset Maintenance,
Utilities for TBM, 5-6
Dataset Names,
CRAY-1A, 2-3
Dataset Names, 2-6
Dataset Organization
on the TBM, 2-5
Dataset to the TBM,
DISPOSE a CRAY-1A, 6-3
Dataset to the IBM 4341,
DISPOSE a CRAY-1A, 6-5
Dataset Written on CDC 7600,
ACQUIRE TBM, 6-4
Datasets,
CDC 7600, 3-7
CRAY-1A JCL Used to Move, 3-2
CRAY-1A, 2-3
CRAY-1A, 3-6
IBM, 2-6
Labeled, 2-2
LOCAL, 2-4
Moving, 3-1
Non-NCAR Labeled, 2-3
PERMANENT, 2-4
TBM, 2-4
Types of, 2-3
Unlabeled, 2-3
Datasets on the TBM,
Storing and Retrieving, 3-6
Datasets on the TBM,
Archiving, 5-3
Datasets to Network Nodes,
Moving, 3-17
Datasets to the DICOMED,
Moving, 3-21
Datasets to the MODCOMP,
Disposing, 3-21
Datasets to your RJE Site via RSCS,
Disposing, 3-21
Deck Setup Examples,
Job, 6-1
Dedicated Reel,
Archival, 5-5
Dedicated Reel,
Joint User, 5-5
Dedicated Reel (MVN), 5-4
Dedicated Reel Use,
Guidelines for, 5-5
Dedicated Reel Users,
Definitions for, 5-4
Dedicated Reels, 5-2
Definitions,
Dataset, 2-2
Technical Descriptions and, 2-1
for Dedicated Reel Users, 5-4
Descend,
Indirect, 5-5
Devices,
File and Record Access on MSS, 2-8
DICOMED,
Moving Datasets to the, 3-21
DIR=ddn, 4-22
Direct (Random) I/O,
BRANIO:, 4-3
Direct Tape,
Examples Using Cray, 3-16
Direct Tape Connection,
Cray, 3-8
Disk,
CRAY-1A, 2-8
Reading a CRAY-1A Tape
to a CRAY-1A, 6-6
DISPOSE
Statement, 3-2
a CRAY-1A Dataset to the TBM, 6-3
a CRAY-1A Dataset to the IBM 4341, 6-5
JCL Statements,
and ACQUIRE, 2-8
Parameters, 3-6
Disposing Datasets to
the MODCOMP, 3-21
Disposing Datasets to your
RJE Site via RSCS, 3-21
Disposition of Dataset, 3-3
DSEG, 4-29
DTLIST, 5-9
Effective use of the TBM, 5-1
Entry Point GBYTES, 4-30
Envelope,
The TBM, 2-4
Error Detection on the TBM, 2-6
Examples,
Job Deck Setup, 6-1
Using Cray Direct Tape, 3-16
Fiche,
Print Character Data from Tape, 3-22
Field Keywords,
TEXT, 3-10
File and Record Access on
MSS Devices, 2-8
File Names,
Special Problems with TBM Dataset, 2-5
IBM, 2-6
Files,
Binary CMS, 3-19
CMS Text and Program, 3-17
Plot or Metacode, 3-22
Storing and Retrieving IBM-CMS, 3-17
FN=, 4-20
Format,
Blocked Record, 2-5
Dataset, 3-3
Other Dataset, 2-7
Segmented Dataset, 2-7
FS=, 4-20
Further Assistance, 5-11
GBYTES,
Entry Point, 4-30
GBYTES-GBYTE & SBYTES-SBYTE, 4-30

ii
Reading a CRAY Tape to a
CRAY Disk, 6-6
Reading and Writing Your VSNs, 5-1
Record Access on MSS Devices,
File and, 2-8
Record Format,
Blocked, 2-5
Record Structure,
TB M Physical, 2-5
Reel,
Archival Dedicated, 5-5
Joint User Dedicated, 5-5
Dedicated, 5-4
Reel Use,
Guidelines for Dedicated, 5-5
Reel Users,
Definitions for Dedicated, 5-4
REFMT, 4-23
Resources,
A Note on Generic, 3-9
Retrieving Datasets on the TBM,
Storing and, 3-6
Retrieving IBM-CMS Files,
Storing and, 3-17
Review,
Command Syntax, 6-1
RJE Site via RSCS,
Disposing Datasets to your, 3-21
Routines,
Data Conversion, 4-19
Input/Output and
Dataset Conversion, 4-1
I/O, 4-2
Routines for Converting 7600
TBM Volumes, 4-19
Routines for the Conversion
of CDC 7600 Data, 4-25
RPTIN, RPTOUT, 4-14
RPTOUT,
RPTIN, 4-14
RSCS,
Disposing Datasets to your
RJE Site via, 3-21
SBYTES-SBYTE,
GBYTES-GBYTE &, 4-30
SCONV, 4-25
Segmented Dataset Formats, 2-7
Serial Name (VSN),
Volume, 5-4
Setup Examples,
Job Deck, 6-1
Special Problems with
TBM Dataset and File Names, 2-5
Staging Jobs on the
CRAY-1A Computers, 3-1
Staging Tapes, 3-8
Statements,
ACQUIRE and DISPOSE JCL, 2-8
Storing and Retrieving
Datasets on the TBM, 3-6
IBM-CMS Files, 3-17
Structure,
TBM Physical Record, 2-5
Subroutine UBLK, 4-9
Subroutine UZBLK, 4-11
Summary Tables, 3-3
Syntax Review,
Command, 6-1
System,
The AMPEX Terabit Memory, 1-3, 2-1
On-line, 5-5
Tables,
Summary, 3-3
Tape,
1/2-inch, 2-8
Examples Using
Cray Direct, 3-16
Log, 5-4
Production (prod), 5-4
Tape on Fiche,
Print Character Data from, 3-22
Tape on the IBM,
Converting 7-track Tape
to 9-track, 6-23
Tape to 9-track Tape on the IBM,
Converting 7-track, 6-23
Tapes,
CRAY 1/2-inch, 1-4
IBM 4341 1/2-inch, 1-4
Staging, 3-8
TBM,
ACQUIRE a CRAY Dataset from, 6-3
ACQUIRE, 6-4
Archiving Datasets on the, 5-3
Binary CMS Files on the, 3-19
Dataset Conventions, 2-4
Dataset Maintenance,
Dataset Organization on the, 2-5
Dataset Written on CDC 7600,
Datasets, 2-4
DISPOSE a CRAY Dataset to the, 6-3
Effective use of the, 5-1
Envelope, 2-4
Error Detection on the, 2-6
Physical Record Structure, 2-5
Plot or Metacode Files from the, 3-22
Special Problems with, 2-5
Storing and Retrieving Datasets on the
TBM Dataset and File Names, 2-5
Staging Jobs on the
CRAY-1A Computers, 3-1
Utilities for, 5-6

iv
TBMCONV, 4-19
TRMVSN, 5-8
Terabit Memory System,
   The AMPEX, 1-3
Terabit Memory System (TBM),
   The AMPEX, 2-1
Terms Used, 3-8
Text and Program Files,
   CMS, 3-17
TEXT Field Keywords, 3-10
The AMPEX Terabit Memory System, 1-3, 2-1
The NCAR Local Network, 1-5
The TBM Envelope, 2-4
Twin CRAY-1A Computers, 1-4
Types of Datasets, 2-3
UBLOK,
   Subroutine, 4-9
Unlabeled Datasets, 2-3
UPKEEP, 5-7
Use, 4-29
Use,
   Guidelines for Dedicated Reel, 5-5
   of the TBM, 5-1
Used,
   Terms, 3-8
   Names and Parameters, 4-3
Used to Move Datasets,
   CRAY-1A JCL, 3-2
User Dedicated Reel,
   Joint, 5-5
Users,
   Definitions for Dedicated Reel, 5-4
Using Cray Direct Tape,
   Examples, 3-16
Utilities for TBM Dataset Maintenance, 5-6
UZBLOK,
   Subroutine, 4-11
Volume Serial Name (VSN), 5-4
Volumes,
   Routines for Converting 7600 TBM, 4-19
VSN,
   Volume Serial Name, 5-4
VSN Moves, 5-5
VSN Purge Policy, 5-3
VSNs,
   Reading and Writing Your, 5-1
VSNs,
   Large, 5-2
Written on CDC 7600,
   ACQUIRE TBM Dataset, 6-4
WRTAPE, IOWAIT,
   RDTAPE, 4-5