MRX/OS Utility Programs

Reference Manual

2200.006-01

MEMOREX

Computer System
Products

May 1973 Edition

This edition is a major revision and obsoletes all previous editions. It documents the MRX/OS utility programs at their level in MRX/OS Release 2.

Technical changes are marked with a vertical bar in the outer margin. Changes due to subsequent releases will be documented in future publications bulletins or revisions.

Requests for copies of Memorex publications should be made to your Memorex representative or to the Memorex branch office serving your locality.

A readers' comments form is provided at the back of this publication. If the form has been removed, comments may be addressed to the Memorex Corporation, Publications Dept., 8941 – 10th Ave. No. (Golden Valley) Minneapolis, Minnesota 55427.

PREFACE

This manual describes the MRX/OS utility programs. The programs are described separately, with each description containing information about program capabilities and required control information. Program descriptions are grouped according to general program type (system service, file-to-file, and Memorex conversion). An appendix is also included to contain program messages, operating information for the stand-alone programs (those that do not run under MRX/OS control), and other supplementary information.

In using this manual, readers should have access to, or be familiar with, the following MRX/OS manuals:

- MRX/OS Control Language Services, Extended Reference
- MRX/OS Control Program and Data Management Services, Basic Reference
- MRX/OS Control Program and Data Management Services, Extended Reference
- MRX/OS Messages

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Memorex utility programs are generalized routines for functions that must be performed frequently in many data processing installations. MRX/OS Control Language statements are used to modify the operation of these programs to meet the needs of individual users. Users can write several control statements, rather than a complete program, to perform a variety of standard operations (a symbolic assembly is not necessary).

PROGRAMS DESCRIBED IN THIS MANUAL

The general categories of Memorex utility programs are: system service, file-to-file, and conversion. System service utilities perform system housekeeping duties. Those described in this manual are:

- Stand-Alone Disc Initialize (with surface analysis and alternate track assignment if necessary) – (UTSSDI)
- Allocate File (UTSSAF)
- Purge File (UTSSPF)
- Catalog and Uncatalog File (UTSSCU)
- Change Volume Serial Number (UTSSCL)
- System Catalog Display (UTSSCD)
- Rebuild Alternate Track (UTSSRT)
- Stand Alone Memory Dump (UTSSMD)
- Stand-Alone Disc-to-Disc Copy (UTSSDD)
- Tape Initialize Program (UTSSTI)
- Stand Alone Disc-to-Tape Dump (UTSSDT)
- Stand Alone Tape-to-Disc Restore (UTSSTD)
- Load Universal Character Set (UTSSUU)

The file-to-file utility program (UTFF) transfers data files from:

- Cards to card, tape, disc, or printer.
- Tape to card, tape, disc, or printer.
- Disc to card, tape, disc, or printer.

The conversion utility programs enable users to convert:

- IBM Disc Volumes to Memorex Disc Volumes (UTCVIM)
- Converted Memorex disc volumes back to IBM Disc Volumes (UTCVMI)
- IBM Tape Files to Memorex Tape Files (UTCVTM)
- Emulated M20 IBM Disc File to Memorex Disc File (UTCVEM)
- IBM M20 Disc Files to Emulated IBM M20 Disc Files (UTDFCU)

Some of the utilities run under control of the Memorex Operating System (MRX/OS), while others are stand-alone programs and run independently of MRX/OS; they include the following:

- Stand-Alone Disc Initialize
- Stand-Alone Disc-to-Disc Copy
- Stand-Alone Memory Dump
- Stand-Alone Disc to Tape Dump
- Stand-Alone Tape to Disc Restore

All of the utilities except the Stand-Alone Memory Dump program run in main memory of the processing unit. The Stand-Alone Memory Dump program runs in control memory.

HARDWARE REQUIREMENTS

The utilities that run under MRX/OS require the minimum MRX/40 or MRX/50 system configuration, and any peripheral equipment specified by the utility program. The stand-alone utilities have individual requirements; all of them, however, require a card reader for program loading. Table 1-1 lists the hardware requirements.

Table 1-1. Hardware Requirements

Program	Minimum Hardware Requirement
Utilities Running Under MRX/OS	One 7200 or 7300 Processing Unit with 16K bytes of main memory* One 3664 Disc Storage Drive UTSSCL, UTCVIM, and UTCVMI require two disc drives. One 1240 Printer-Keyboard (console) One 8010 Card Reader or 8025 Card Read/Punch One 5120 Printer UTCVTM requires two 3237 tape drives.
Disc Initialize	One 7200 or 7300 Processing Unit with 16K bytes of main memory One 3664 Disc Storage Drive One 1240 Printer-Keyboard (console) One 8010 Card Reader or 8025 Card Read/Punch
Disc-to-Disc Copy	One 7200 or 7300 Processing Unit with 16K bytes of main memory Two 3664 Disc Storage Drives One 1240 Printer-Keyboard (console) One 8010 Card Reader or 8025 Card Read/Punch
Stand-Alone Memory Dump	One 7200 or 7300 Processing Unit with 4K words of control memory One 8010 Card Reader or 8025 Card Read/Punch One 5120 Printer or other standard EBCDIC character-set line printer
Disc to Tape Dump and Tape to Disc Restore	One 7200 or 7300 Processing Unit with 16K bytes of main memory One 3664 Disc Storage Drive One 3237 Tape Drive One 1240 Printer-Keyboard (console) One 8010 Card Reader or 8025 Card Read/Punch

*If the Resident Extension System version of MRX/OS is used, the minimum amount of memory required for the UTCVIM, UTCVMI, and UTDFCU programs is 24K bytes.

PROGRAMMING SYSTEM REQUIREMENTS

All of the utilities that run under MRX/OS operate under either version of MRX/OS (Minimum Operating System or Resident Extension System). All of the programs, except UTSSCD, UTSSCU, UTCVIM, UTCVMI, UTFF and UTDFCU execute in a 4K byte partition. UTDFCU executes in a 12K partition and the other exceptions operate in an 8K partition.

CONTROL LANGUAGE

MRX/OS Control Language statements (CLS) provide automatic job to job transition during a run, pass parameters to utility programs, control card input to utility programs, direct data management, and assign the required system resources. Memorex utility programs running in conjunction with the operating system use the standard Control Language statements. Parameters for the parameter statements required by each utility program are included in individual utility program descriptions.

The utility program parameters follow the KEYWORD=data format of other Memorex system parameters and are specified in //PAR statements. Keyword parameters included in the same //PAR statement are separated by commas; every parameter except the last parameter of the statement is followed by a comma. A statement requires multiple cards, every parameter except the last parameter on the last //PAR card is followed by a comma.

The standard sequence of control statements used with Memorex utility programs is:

1. //JOB statement
2. //EXECUTE statement
3. //DEFINE statement(s)
4. //PAR statement(s)
5. //EOJ statement

See Appendix B in this manual for a summary of the control statements; detailed descriptions of control statements can be found in the **MRX/OS Control Language Services, Extended Reference** manual.

2. SYSTEM SERVICE UTILITY PROGRAMS

System service utility programs perform maintenance functions necessary for the support of the operating system. The programs are:

- Stand-Alone Disc Initialize
- Allocate File
- Purge File
- Catalog and Uncatalog
- Change Volume Serial Number
- System Catalog Display
- Rebuild Track
- Stand-Alone Memory Dump
- Stand-Alone Disc-to-Disc Copy
- Tape Initialize
- Stand-Alone Disc-to-Tape Dump
- Stand-Alone Tape-to-Disc Restore
- Load Universal Character Set
- Disc File Conversion Unit Copy

DISC INITIALIZE PROGRAM – UTSSDI

The Disc Initialize program does not run under control of MRX/OS; it executes as a stand-alone program. The user communicates with the program through the operator's console (see *UTSSDI Parameters* in following text). Procedures for loading and running the program are discussed under *Disc Initialize Procedure* in Appendix D.

UTSSDI PROGRAM CAPABILITIES

The Disc Initialize utility program prepares one work or system-resident disc pack for use on a Memorex disc drive. This program effects:

1. Label checking
2. Writing home addresses
3. Surface analysis – testing track surfaces
4. Alternate track assignment
5. Label and catalog generation
6. Cylinder zero updating

Disc pack cylinder 0 is reserved for system use. This utility program writes a home address preceding the first record of every track and a track descriptor record as the first record of every track (record 0). The program also writes a volume device label (cylinder 0, head 0, record 3) and updates the alternate track assignment table (cylinder 0, head 0, record 4).

LABEL CHECKING

The Disc Initialize program checks for an existing device label (see Appendix A). If a label exists, it is printed on the operator's console. The operator can continue processing or terminate the job (see *UTSSDI Operator Information* later in this discussion for more information).

HOME ADDRESS

The home address field is written at the beginning (preceding record 0) of each track. It defines the track address and condition. The home address field contains a cylinder number and head number, and is preceded by a flag byte (Figure 2-1).

Home address field	Flag	Cylinder number	Head number	
	0	1 2	3 4	Bytes

Flag byte:

Bit	Description	Content
0-5	Zeros	000000
6	Track condition	0 — operative track 1 — defective track
7	Track use	0 — primary track 1 — alternate track

Figure 2-1. Home Address Field

SURFACE ANALYSIS

The surface analysis routine tests the condition of track surfaces by writing test patterns on the tracks. If a track is defective, bit 6 of the home address flag byte is set to 1 and an alternate track is assigned for the defective track. If a track is not defective, the track descriptor record (record 0) is written on the track.

A track description error (inability to write the track descriptor record on the track) results in job termination when surface analysis is completed. The track descriptor record has a nine byte count field and an eight byte data field (Figure 2-2).

ALTERNATE TRACK ASSIGNMENT

When the Disc Initialize program detects a defective track during surface analysis, it assigns an alternate track by switching the track address fields (bytes 1-4 of track descriptor record count field) of the defective and alternate tracks. When the supply of operative alternate tracks is exhausted, the job terminates.

LABEL AND CATALOG GENERATION

The Disc Initialize program generates a pack catalog (PCAT) on each pack and a central catalog (CCAT) on system resident packs. An eighty-byte device label is written on cylinder 0, head 0, record 3*.

*See **MRX/OS Control Program and Data Management Services, Reference** manuals for detailed descriptions and formats of these catalogs and the device label.

Count field	Flag	Cylinder number	Head number	Record number	Key length	Data length

0 1 2 3 4 5 6 7 8 Bytes

Count field flag:

Bit	Description	Content
0	Odd/even indicator	0 – even number of records on track 1 – odd number of records on track
1-5	Zeros	000000
6	Track condition	0 – operative track 1 – defective track
7	Track use	0 – primary track 1 – alternate track

Data field	Cylinder number	Head number	Record number	Number bytes remaining in track	Binary zeros

0 1 2 3 4 5 6 7 Bytes

Figure 2-2. Track Descriptor Record

CYLINDER 0 UPDATING

If cylinder 0 control memory data is to be updated, a cylinder 0 update deck must be in the card reader. See Appendix C for this deck structure. If cylinder 0 is not to be updated, a data delimiter card (/*) must be in the card reader. See the CYL0 keyword under *UTSSDI Parameters* for more information on cylinder 0 updating.

UTSSDI PARAMETERS

The program requests the following parameters from the operator by printing the keywords on the operator's console. The operator enters the appropriate parameter for the keyword (see *UTSSDI Operator Information* in following text). If less than the maximum number of parameter characters are entered, the operator uses a carriage return after the last character to proceed to the next parameter. Otherwise, a carriage return is not used.

Keyword	Parameter	Description
DRIVE=	n	A 1-digit drive number (0-7) of the disc drive that the pack to be initialized is mounted on. Default value is 1.

Keyword	Parameter	Description
CARDDEV=	nn	A two-character card reader device address. Valid entries are 04, 0C, and 24. Default is 0C.
CYL0=	YES NO	YES specifies that the only initialization step the program will perform is updating cylinder 0 (control memory data). The cylinder 0 update deck (Appendix C) must be in the card reader. NO specifies that a complete initialization is to be performed. Cylinder 0 is updated if a SYSRES pack is being built and the update deck (Appendix C) is in the card reader. Otherwise, two data delimiter cards (/*) must be in the reader. Default is NO.
VOL=	volid	A 1-6 character (alphanumerics, $, and -) volume identifier to be entered into the disc pack device label. If less than 6 characters are specified, they are left-justified and blank-filled. All preceding blanks or embedded blanks are invalid. Required entry.
OWNER=	ownerid	A 1-54 character owner identification. Default value is blanks.
PCAT=	ccc,n	A decimal specification of the starting cylinder address (ccc) and the number of cylinders (n) for pack catalog allocation. If PCAT= is specified, both sub-parameters must be supplied. Leading zeros are not required; n must be greater than zero, and ccc+n must be less than 202. Default value is 1, 1.
CCAT=	ccc,nn	A decimal specification of the starting cylinder address (ccc) and the number of cylinders (nn) for central catalog allocation. If no CCAT is generated, cylinder 0 is not updated. For system resident packs, placing PCAT on cylinders 1 and 2 and CCAT on cylinder 3 provides more efficient performance. If CCAT= is specified, both

Keyword	Parameter	Description
		sub-parameters must be supplied. Leading zeros are not required; nn must be greater than zero; and ccc+nn must be greater than zero. Default is no central catalog.
PASS=	n	A 1-character specification of the number of surface test passes. If PASS=0, no surface test is performed. Default value is 1.
PATTERN=	hh	A 2-character hexadecimal representation of the byte pattern to be written during the surface test. This parameter is not requested if PASS=0 is specified. Default value is E5.
ALTADR=	ccc	A 3-digit decimal specification of the alternate track area starting cylinder address. (Leading zeros must be specified.) The next three cylinders are alternate track area (or as many cylinders as are available if this is the end of the disc). Default value is 200.

UTSSDI OPERATOR INFORMATION

The Disc Initialize program operates independently of MRX/OS. The operator controls program execution through the operator's console (see Appendix D for detailed procedures). When the program is loaded, it prints a header message and begins printing a series of parameter request messages. For example:

```
DISC INITIALIZE REV mm-dd-yy

PARAMETERS
DRIVE=
```

After each request for a parameter, the operator enters the required parameter or pushes the carriage return key. A carriage return causes the default for the parameter to be used. If less than the maximum number of parameter characters are entered, the operator uses a carriage return after the last character to proceed to the next parameter. Otherwise, a carriage return is not used.

The Disc Initialize program validates every operator entry. If the parameter is valid, the program proceeds to the next parameter request. If the parameter is not acceptable, the program displays the message INVALID ENTRY and repeats the parameter request. Console parameter requests continue until every parameter has been supplied or until the operator enters ETX. The default values are then assumed for the remaining parameters.

When the utility has completed execution, an end-of-program message is printed on the console stating volume number, drive number, and alternate track assignments. The utility begins requesting parameters again; another volume can be initialized. If the Disc Initialize program has not been restarted, parameter default values are equal to the parameter entries for the volume just initialized.

Disc Initialize error messages are listed in Appendix E. Some messages request information from the operator, as shown in the following example:

```
DEVICE LABEL EXISTS WITH VOLUME IDENT=DDDDDD
CONTINUE?
```

In this example, a volume identification of DDDDDD indicates that an abnormal termination on a previous initialization occurred. The operator could reply with a Y or YES to re-initialize the pack. (The operator would not press the INT key before responding.)

Another example of a message requesting information is as follows:

```
CARD READ FAILURE
RETRY?
```

In this case, required cards are not in the card reader. If CYL0=NO, for example, either the cylinder 0 update deck or a data delimiter card (/* in columns 1 and 2) could be missing from the reader. The operator puts the required cards in the reader, presses START on the reader, and replies with a Y or YES to continue. (The operator does not press INT before responding.)

ALLOCATE FILE PROGRAM – UTSSAF

The Allocate File program runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTSSAF PROGRAM CAPABILITIES

The Allocate File utility program allocates disc space for a new pair of permanent user files sharing cylinder allocation (Control Language Services enables the allocation of all other files). A specific volume for file allocation can be named in the //DEFINE statement. For uncatalogued files, the //DEFINE statement and an ID=YES parameter (see *UTSSAF Parameters* in following text) in a //PAR statement are required. For cataloged, if the ID=YES parameter is omitted, the file pair is allocated on a shared device.

UTSSAF PARAMETERS

//PAR statements are required for the Allocate File program. The parameters required in the statements depend upon the file organization of the files being allocated.

For relative or sequential files, the following parameters are used:

Keyword	Parameter	Description
ORG=	{S R}	S – sequential file organization R – relative file organization Default is S.

For sequential files only, the following parameters are used:

Keyword	Parameter	Description
FILENAME=	filename	A 1-17 character EBCDIC file label filename. Required.
MSC=	msc	A 1-4 byte modification security code to be assigned to this file. Default is no msc.
VER=	{YES NO}	YES – All writes to this file will be verified. NO – No write verification done on this file. Default is NO.

Keyword	Parameter	Description
BLCT=	n	Number of blocks – maximum value is 2^{24}-1. Required.
BLKSIZ=	n	The number of 8-bit bytes per block. If the records have a common stored data format header, the byte count of this header (4 bytes) must be included in the block size. Range of values is 18-7294 bytes. Required.
ID=	YES	YES – The user has provided a //DEFINE card containing the specific volume identifiers for the file-pair allocation. NO – Allocation of cataloged files is to shared device. Default is NO.
CAT=	NO YES	NO – File not cataloged in central catalog; specifies volume allocation must be made (ID=YES and a //DEFINE statement must be included). YES – File is cataloged. Default is YES.
FIL2=	paired filename	A 1-17 character EBCDIC name of the file sharing cylinder allocation with the file specified by the FILENAME parameter. Required parameter.

The BLCT and BLKSIZ parameters determine the total amount of space allocated. (This amount is rounded up to an even number of tracks). The allocated space is then divided between the two files:

1. The FIL2 (upper) file is written on the even numbered tracks (tracks 0, 2, 4, . . .).

2. The FILENAME (primary or lower) file is written on the odd numbered tracks (tracks 1, 3, 5, . . .).

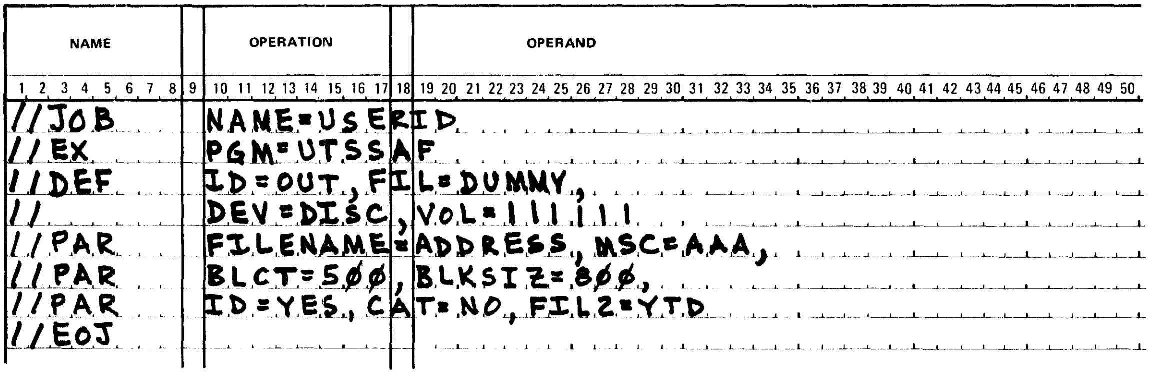
UTSSAF REQUIRED CONTROL LANGUAGE

The Allocate File utility program requires the standard //JOB and //EX statements. A //DEF statement with ID=OUT, FIL=DUMMY, and no other options is required for specific volume allocation and for uncataloged files. //PAR statements contain the UTSSAF parameters.

UTSSAF EXAMPLES

EXAMPLE 1

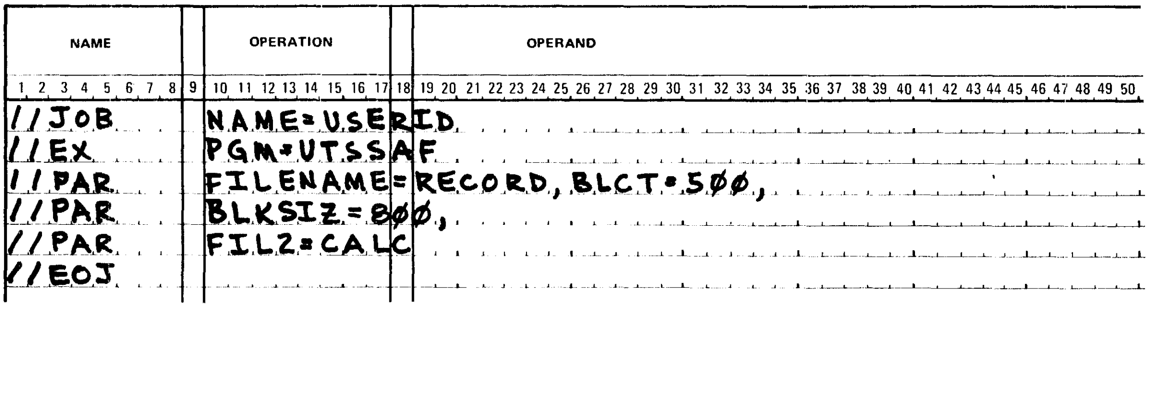
The following example illustrates allocating a pair of non-cataloged files, ADDRESS and YTD, to disc volume 111111. The modification security code is to be AAA. The file identifier is specified in the //DEF statement. The block size is 800 and the block count is 500:



```
NAME      OPERATION  OPERAND
1 2 3 4 5 6 7 8 | 9 | 10 11 12 13 14 15 16 17 | 18 | 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB    NAME=USERID
//EX     PGM=UTSSAF
//DEF    ID=OUT,FIL=DUMMY,
//       DEV=DISC,VOL=111111
//PAR    FILENAME=ADDRESS,MSC=AAA,
//PAR    BLCT=5ØØ,BLKSIZ=8ØØ,
//PAR    ID=YES,CAT=NO,FIL2=YTD
//EOJ
```

EXAMPLE 2

The following example illustrates allocating a pair of cataloged files, RECORD and CALC, to shared devices:



```
NAME      OPERATION  OPERAND
1 2 3 4 5 6 7 8 | 9 | 10 11 12 13 14 15 16 17 | 18 | 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB    NAME=USERID
//EX     PGM=UTSSAF
//PAR    FILENAME=RECORD,BLCT=5ØØ,
//PAR    BLKSIZ=8ØØ,
//PAR    FIL2=CALC
//EOJ
```

PURGE FILE PROGRAM – UTSSPF

The Purge File utility program runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTSSPF PROGRAM CAPABILITIES

The program deletes a file from the system. Two files sharing common allocation (paired files) are both deleted if one of the files is deleted. This program can delete scratch or temporary files, but under normal operating conditions, Control Language Services performs this function at the end of a job step (scratch files) or at the end of a job (temporary files).

UTSSPF PARAMETER

The Purge File program has one keyword parameter that is entered in a //PAR statement.

Keyword	Parameter	Description
PURGE=	(filename,msc,NO,P)	The filename entry is the required 1-17 character name given to the file to be purged when it was allocated. The msc entry is a 1-4 character modification security code in the file label. Required if the MSC field is not blank. NO indicates file not listed in central catalog. If not specified, the central catalog is searched for filename match. P indicates file was paired when allocated. Required if filename is a paired file.

Commas must be substituted for non-specified parameters; for example:

```
PURGE=(FILE1, , ,P)
```

UTSSPF REQUIRED CONTROL STATEMENTS

The Purge File utility program requires the standard //JOB and //EX statements. When purging a scratch or temporary file, the //JOB statement name must be the jobname of the program that created the file to be purged.

A //DEF with ID=OUT, FIL=DUMMY, and no other options is required if the specified file resides on non-system resident packs or for uncataloged files to specify the appropriate volume. One //PAR statement is required for each file or pair of files to be purged. Scratch and temporary files are an exception to this requirement; a //PAR statement is optional (see Example 4).

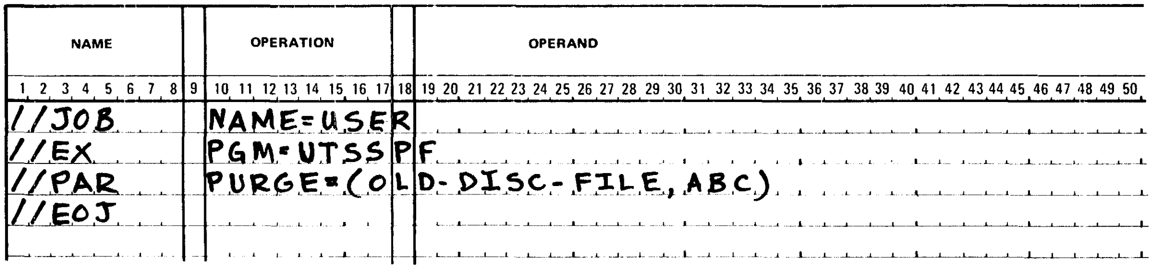
If multiple files are to be purged in one execution of the Purge Utility program:

- All the files must reside on the same disc pack; only one volume ID can be specified in the //DEFINE statement.
- PURGE must appear once and only once in each //PAR statement.

UTSSPF EXAMPLES

EXAMPLE 1

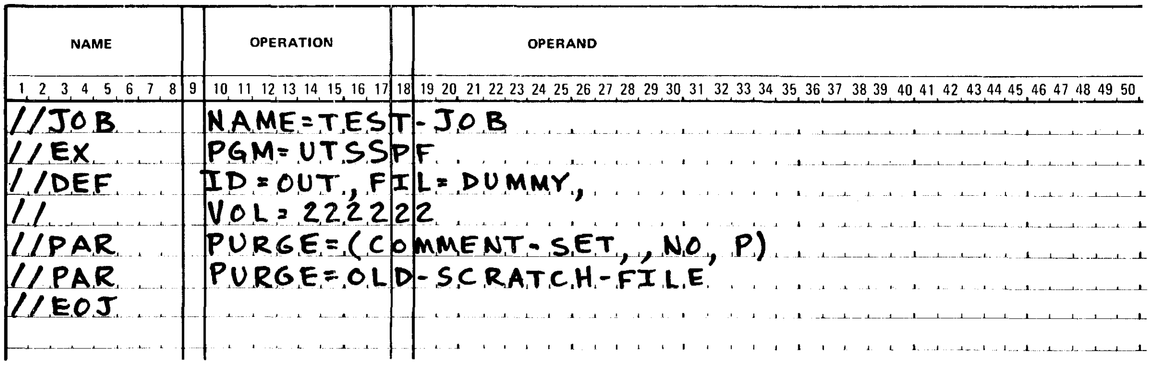
This example purges a single non-paired cataloged file named OLD-DISC-FILE with the modification security code of ABC from the system-resident volume 111111:



```
NAME      OPERATION  OPERAND
//JOB      NAME=USER
//EX       PGM=UTSSPF
//PAR      PURGE=(OLD-DISC-FILE,ABC)
//EOJ
```

EXAMPLE 2

This example purges three files (two of which are paired) on the non-system-resident volume 222222:



```
NAME      OPERATION  OPERAND
//JOB      NAME=TEST-JOB
//EX       PGM=UTSSPF
//DEF      ID=OUT,FIL=DUMMY,
//         VOL=222222
//PAR      PURGE=(COMMENT-SET,,NO,P)
//PAR      PURGE=OLD-SCRATCH-FILE
//EOJ
```

The two permanent files, UPDATE-SET and COMMENT-SET (the primary file), were paired but not cataloged when they were allocated. The file OLD-SCRATCH-FILE was cataloged and assigned the qualifier TEST-JOB when it was allocated. A modification security code was not assigned to any of these files when they were allocated.

EXAMPLE 3

This example purges a three-volume cataloged file with no modification security code assigned. The file is named LARGE-DATA-SET and resides on the non-system-resident volumes 111111, 222222, and 333333, mounted concurrently on three separate disc drives:

```
NAME     OPERATION OPERAND
//JOB    NAME=USERS
//EX     PGM=UTSSPF
//DEF    ID=OUT,FIL=DUMMY,DEV=(DISC,3),
//       VOL=(111111,222222,333333)
//PAR    PURGE=LARGE-DATA-SET
//EOJ
```

EXAMPLE 4

This example purges a three-volume cataloged file with no modification security code assigned. The file is named LARGE-DATA-SET and resides on the non-system-resident volumes 111111, 222222, and 333333, which will be mounted sequentially on the same drive.

```
NAME     OPERATION OPERAND
//JOB    NAME=USERS
//EX     PGM=UTSSPF
//DEF    ID=OUT,FIL=DUMMY,DEV=DISC,
//       VOL=(111111,222222,333333)
//PAR    PURGE=LARGE-DATA-SET
//EOJ
```

EXAMPLE 5 – USING UTSSPF TO PURGE A SCRATCH OR TEMPORARY FILE

To purge a scratch or temporary file left by an interrupted job, the //JOB statement job name must be the name of the program that created the file to be purged. If the interrupted job creating the scratch or temporary file included the following control language:

NAME		OPERATION		OPERAND
1 2 3 4 5 6 7 8	9	10 11 12 13 14 15 16 17	18	19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB		NAME=TRY	O	UT
//EX		PGM=BADP	G	M
//DEF		ID=SCRAT	C	H, FIL=SCRATCH-FIL, NUMBER=2ØØ,....
//EOJ				

Then the following control language could be used to run the Purge File utility program (the //PAR statement is optional):

NAME		OPERATION		OPERAND
1 2 3 4 5 6 7 8	9	10 11 12 13 14 15 16 17	18	19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB		NAME=TRY	O	UT
//EX		PGM=UTSS	P	F
//PAR		PURGE=SC	R	ATCH-FIL
//EOJ				

CATALOG AND UNCATALOG PROGRAM – UTSSCU

This utility program catalogs and uncatalogs files. Cataloging a file is entering information about the file in the central catalog and setting the file's CCAT bit in the attribute element of the pack catalog on each volume on which the file resides. This file entry includes file name, file attributes, and a pointer to the file entry in the pack catalog(s). Uncataloging a file is removing the file's entry from the central catalog and resetting the CCAT bit in the attribute element of each pack catalog.

The user can specify cataloging and uncataloging in the same job step; the uncataloging is done first. If the user specifies either one or both of a pair of file names, both of the files will be cataloged or uncataloged. Other access to the central and pack catalogs is prevented during cataloging and uncataloging. The total number of files that can be cataloged and uncataloged in one job step is limited by the number of //DEF statements that can be included in a job step and by the number of volumes which must be mounted during step initiation.

//DEF statements specify the file names and volume identifiers of the files to be cataloged and uncataloged. The volume identifiers must be listed in the same order as they were listed in the //DEF statements used to allocate the file. If a file index was allocated to a different volume than the file, the file index volume must be specified in the //DEF statement (IVO keyword).

UTSSCU REQUIRED CONTROL LANGUAGE

The Catalog-Uncatalog program requires the standard //JOB and //EX statements. A //DEF statement is required to identify the central catalog volume (the system resident volume for uncataloging), and one //DEF statement is required for each file to be cataloged or uncataloged. An //EOJ statement terminates the job step. Table 2-1 lists the //DEF statement parameters applicable to the Catalog-Uncatalog program.

Table 2-1. //DEF Statement Parameters Applicable to UTSSCU Program

Parameters for Identifying Central Catalog	Parameters for File to be Cataloged	Parameters for File to be Uncataloged
ID=IN	ID=CTLGnn*	ID=UNCTLGnn*
FIL=$CCAT	FIL=filename	FIL=filename
DEV=DISC	$\text{DEV=}\left\{\begin{matrix}\text{DISC}\\\text{(DISC,n)}\end{matrix}\right\}^{**}$	$\text{DEV=}\left\{\begin{matrix}\text{DISC}\\\text{(DISC,n)}\end{matrix}\right\}^{**}$
STA=(P,O)	STA=(P,O)	STA=(P,O)
VOL=volid	$\text{ORG=}\left\{\begin{matrix}\text{S}\\\text{R}\\\text{I}\end{matrix}\right\}$	$\text{ORG=}\left\{\begin{matrix}\text{S}\\\text{R}\\\text{I}\end{matrix}\right\}$
	MSC=code	MSC=code
	$\text{VOL=}\left\{\begin{matrix}\text{volid}\\\text{(volid,...,volid)}\end{matrix}\right\}$	$\text{VOL=}\left\{\begin{matrix}\text{volid}\\\text{(volid,...,volid)}\end{matrix}\right\}^{\dagger}$

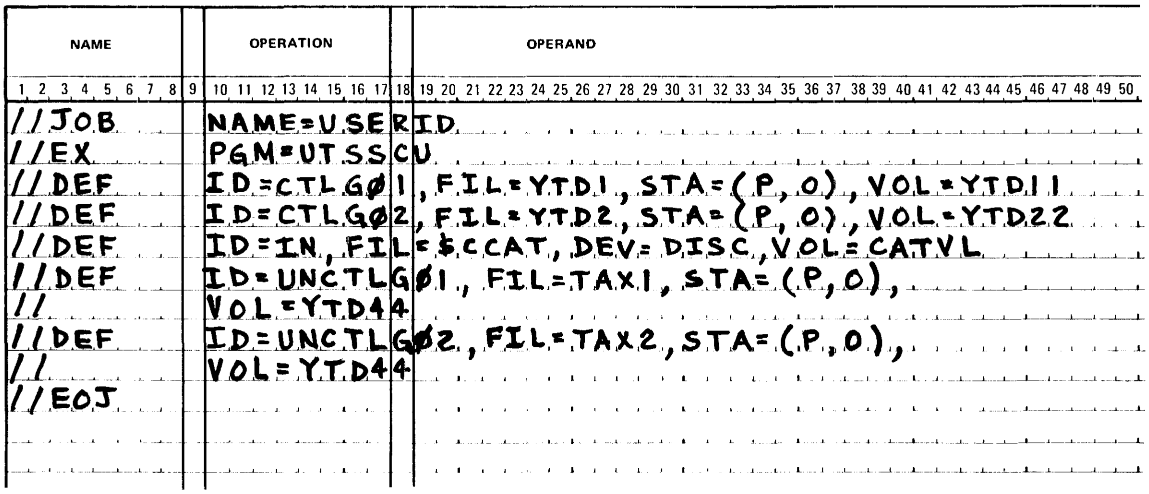
*nn are consecutive numbers beginning at 01; 01 for the first file to be cataloged, 01 for the first file to be uncataloged, 02 for the second file to be cataloged, etc.

**The number of discs must be specified if there is more than one volume identifier and a volume identifier refers to a shared device, or if the file organization is specified as relative or indexed.

†If an uncataloging //DEF statement contains a volume identifier that does not match an entry in the central catalog, the file will not be uncataloged.

UTSSCU EXAMPLE

The following Control Language statements catalog the two files YTD1 and YTD2 and uncatalog the two files TAX1 and TAX2:



```
//JOB     NAME=USERID
//EX      PGM=UTSSCU
//DEF     ID=CTLGØ1,FIL=YTD1,STA=(P,0),VOL=YTD11
//DEF     ID=CTLGØ2,FIL=YTD2,STA=(P,0),VOL=YTD22
//DEF     ID=IN,FIL=$CCAT,DEV=DISC,VOL=CATVL
//DEF     ID=UNCTLGØ1,FIL=TAX1,STA=(P,0),
//        VOL=YTD444
//DEF     ID=UNCTLGØ2,FIL=TAX2,STA=(P,0),
//        VOL=YTD444
//EOJ
```

CHANGE VOLUME SERIAL NUMBER PROGRAM – UTSSCL

The Change Volume Serial Number program runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTSSCL PROGRAM CAPABILITIES

The Change Volume Serial Number utility program changes the volume serial number and, optionally, the owner identifier and/or the pack status in a disc pack device label. (See Appendix A for standard label formats.) The volume whose identifier is to be changed must not be a shared disc or the SYSRES. The central catalog and pack catalogs are not updated.

UTSSCL PARAMETERS

The following UTSSCL parameters are specified in //PAR statements.

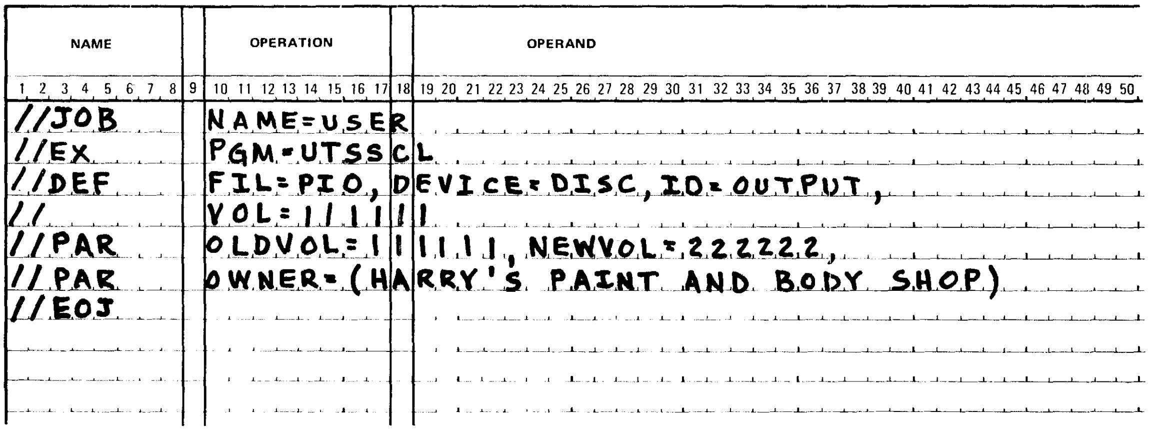
Keyword	Parameter	Description
OLDVOL=	volid	A 1-6 character volume identifier in a disc pack device label. Required parameter.
NEWVOL=	volid	The 1-6 character (alphanumeric and $) volume identifier to be entered in the device label. If less than 6 characters are specified, they are left justified and blank-filled. Required parameter.
OWNER=	ownerid	A 1-54 character identification. If the owner identifier contains blanks, it must be enclosed within parentheses. Default is no change to owner identifier field.
PKSTAT=	$\left\{\begin{matrix} \text{U} \\ \text{R} \\ \text{L} \end{matrix}\right\}$	U – unrestricted volume – any allocate allowed. R – restricted volume – no file allocation allowed from control. L – locked volume – no new file allocations allowed on this pack. Default is no change to pack status field.

UTSSCL REQUIRED CONTROL LANGUAGE

The Change Volume Serial Number utility program requires the standard //JOB and //EX statements and a //DEF statement for the volume whose label is to be changed. The volume must be on a non-shared device.

UTSSCL EXAMPLE

The following is an example of the job control and utility parameter statements necessary to change a volume serial number from 111111 to 222222 and to insert the volume identifier, HARRY'S PAINT AND BODY SHOP, into the device label:



```
NAME      OPERATION         OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

//JOB     NAME=USER
//EX      PGM=UTSSCL
//DEF     FIL=PIO,DEVICE=DISC,IO=OUTPUT,
//        VOL=111111
//PAR     OLDVOL=111111,NEWVOL=222222,
//PAR     OWNER=(HARRY'S PAINT AND BODY SHOP)
//EOJ
```

SYSTEM CATALOG DISPLAY PROGRAM – UTSSCD

The System Catalog Display utility program runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program. There are *no* //PAR statements associated with the UTSSCD program.

UTSSCD PROGRAM CAPABILITIES

The System Catalog Display utility program provides for listing either the pack catalog(s) or the central catalog or both.

The pack catalog display is a detailed breakdown of file attributes and the location and size of various segments of each file on the pack. The pack catalog display is followed by a listing of the number of available tracks on the pack and a detailed breakdown of the starting address and the number of continuous tracks in each block of available space. The central catalog display is a detailed breakdown of file attributes and the volume identification of each cataloged file.

UTSSCD REQUIRED CONTROL LANGUAGE

The UTSSCD utility program requires the standard //JOB and //EX statements and a //DEF statement with ID=PRINTER for the printer output listing. One //DEF statement is required for each volume whose pack catalog is to be listed. One //DEF statement is required to list a central catalog. If a central catalog resides on more than one volume, UTSSCD must be executed once for each volume on which it resides. Table 2-2 outlines the //DEF statements used for the System Catalog Display program.

Table 2-2. //DEF Statement Parameters Applicable to UTSSCD Program

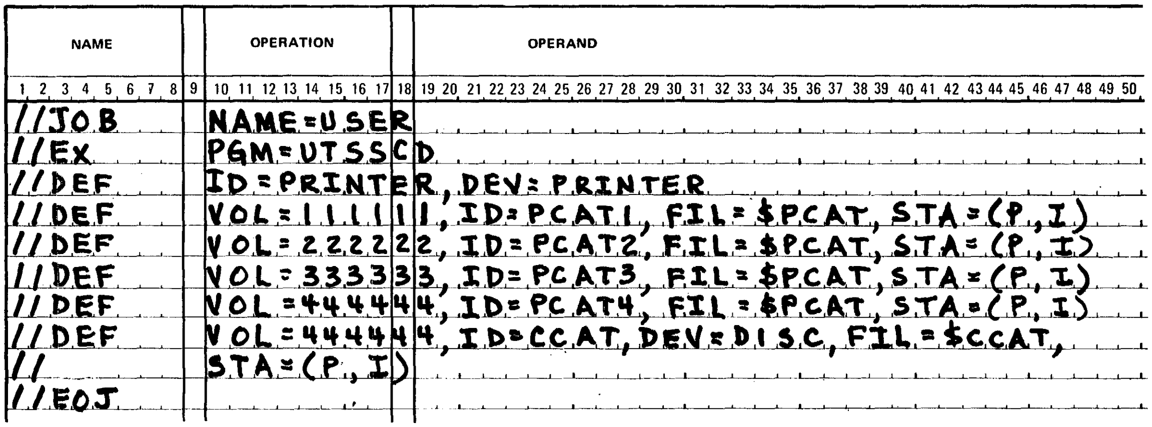
Parameters for Printer Output (Required for Both Catalog Displays)	Parameters for Pack Catalog(s) Display	Parameters for Central Catalog Display
ID=PRINTER	ID=PCATx*	ID=CCAT
DEV=PRINTER	FIL=$PCAT	FIL=$CCAT
	VOL=volid	VOL=volid
	STA=(P,I)	STA=(P,I)

*x is a one-digit number from 1-7. Values of x in multiple //DEF cards must be consecutive integers. The first value must always be 1, as in the following example:

```
//DEF    ID=PCAT1...
//DEF    ID=PCAT2...
//DEF    ID=PCAT3...
```

UTSSCD EXAMPLE

The following is a list of Control Language statements required to list the pack catalog from volumes 111111, 222222, and 333333 and to list both the pack and central catalogs from volume 444444.



```
NAME      OPERATION OPERAND
1       9 10      18 19                                50
//JOB    NAME=USER
//EX     PGM=UTSSCD
//DEF    ID=PRINTER,DEV=PRINTER
//DEF    VOL=111111,ID=PCAT1,FIL=$PCAT,STA=(P,I)
//DEF    VOL=222222,ID=PCAT2,FIL=$PCAT,STA=(P,I)
//DEF    VOL=333333,ID=PCAT3,FIL=$PCAT,STA=(P,I)
//DEF    VOL=444444,ID=PCAT4,FIL=$PCAT,STA=(P,I)
//DEF    VOL=444444,ID=CCAT,DEV=DISC,FIL=$CCAT,
//       STA=(P,I)
//EOJ
```

UTSSCD LISTINGS

The UTSSCD program prints certain standard headings for the catalog displays. These headings are defined in the following text. An example of a catalog display is given in Figure 2-3.

HEADINGS COMMON TO THE PACK CATALOG DISPLAY AND THE CENTRAL CATALOG DISPLAY

Heading	Entry Description
FILE NAME AND QUALIFIER	File name and file name qualifier if there is one
MSC	YES if a modification security code exists NO if a modification security code does not exist N/A where msc does not apply
CRDATE	Creation dates (YY.DDD) YY is the last 2 digits of the year. DDD is the Julian day of the year.
EXDATE	Expiration
STAT	File status: PERM – permanent SCRA – scratch TEMP – temporary WORK – work
TYP	File type (organization): GEN – general (unknown to system) REL – relative SEQ – sequential IND – indexed file data INF – indexed file index ??? – unassigned code found in catalog
VER	YES – write verify specified NO – no write verify

Heading	Entry Description
LBLOK	Block size (bytes) (decimal)
LRECD	Record length (bytes) (decimal)
LKY	Key length (indexed files only) (decimal)
RKP	Relative key position (indexed files only) (decimal)
PF & I	Paired file flag and indicator if PF entry is 0, the indicator is blank (i.e., the file is not paired) If PF=1 entry is 1, the file is paired and the indicator will be: U for an Upper file L for a Lower (primary) file
HIBLOK	Highest block written (relative block number for last block written in the file) (decimal)

HEADINGS EXCLUSIVE TO THE PACK CATALOG DISPLAY

Heading	Entry Description
BC	Boundary on cylinder: 1 if yes, 0 if no
LV	Last volume written flag: 1 if this is the last volume in this file 0 if this is not the last
TSEG	Total number of segments occupied by this file on this pack

The following fields are repeated for files containing more than one segment on a pack:

SEQ	Segment sequence number
CCCHH	Cylinder and head (in decimal) of the starting address of this segment.

Heading	Entry Description
NTRK	Number of tracks in this segment.
NBLKS	Total number of blocks the segment can contain (decimal)

Entries STARTING CCCHH and NUMBER OF TRACKS, in the map of available space are similar in meaning respectively to CCCHH and NTRK. They, however, represent available space on the disc pack.

HEADINGS EXCLUSIVE TO THE CENTRAL CATALOG DISPLAY

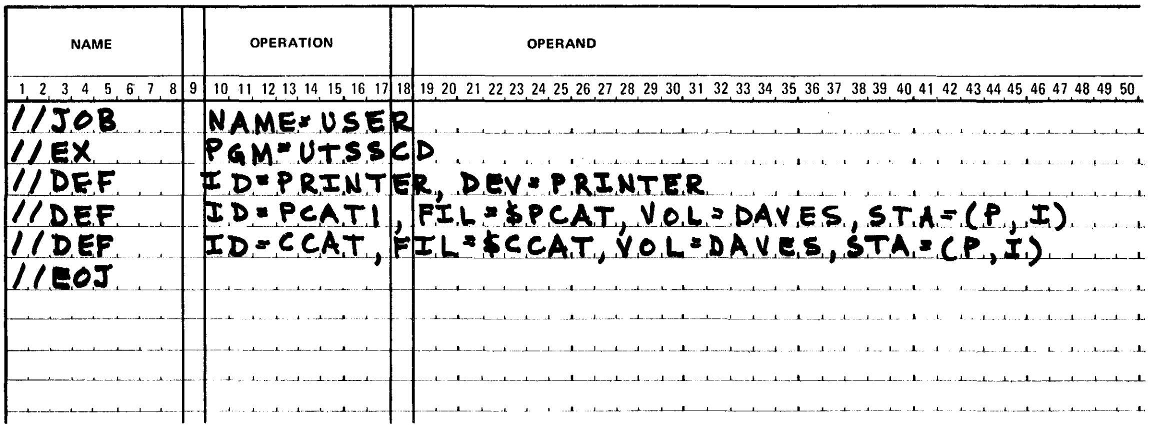
Heading	Entry Description
OFFSET	Relative files – offset (hexadecimal) Sequential files – relative block number of the first block of the highest volume written Information files -- bytes 0-1 count of allocated directory blocks bytes 2-3 count of index blocks
TSEGS	Total number of segments allocated to the file
TVOLS	Total number of volumes allocated to the file

The following field is repeated if the file resides on more than one volume.

VOLUME	Volume identifier

SAMPLE DISPLAY

Figure 2-3 shows the Control Language statements for a pack and central catalog; the resulting pack catalog display is shown in Figure 2-4, the central catalog display in Figure 2-5. The headings on the listing are defined in preceding text. In this example, both the pack and central catalog are on a volume identified as DAVES.



```
  NAME       OPERATION        OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB    NAME=USER
//EX     PGM=UTSSCD
//DEF    ID=PRINTER,DEV=PRINTER
//DEF    ID=PCAT1,FIL=$PCAT,VOL=DAVES,STA=(P,I)
//DEF    ID=CCAT,FIL=$CCAT,VOL=DAVES,STA=(P,I)
//EOJ
```

Figure 2-3. Control Language for Pack and Central Catalog Display

```
*** MEMOREX * SYSTEM * CATALOG * DISPLAY ***    DATE. . .APR 13, 1973    TIME. . .00:31:44.    PAGE. . . 1

VOLUME DAVES  ***PACK CATALOG***

FILE NAME AND QUALIFIER  MSC  CRDATE  EXDATE  STAT  TYP  VER  LBLOK  LRECD  LKY  RKP  PF&I  HIBLOK  BC  LV  TSEG  SEQ  CCCHH  NTRK  NBLKS

$SRTOBJLIB.              NO   00.000  00.000  PERM  SEQ  NO     256    256    0    0  0        863   0   1     1    1   93 9    60   1200
$PCAT                    NO   00.000  00.000  PERM  GEN  NO     128    128    0    0  0        620   0   1     1    1    1 0    20    620
$CCAT                    NO   00.000  00.000  PERM  GEN  NO     128    128    0    0  0        620   0   1     1    1    2 0    20    620
$SYSOBJLIB.              NO   00.000  00.000  PERM  SEQ  NO     256    256    0    0  0       1869   0   1     1    1   46 0   200   4000
JMDATA.                  YES  00.000  00.000  PERM  SEQ  YES    256    256    0    0  0        137   0   1     1    1    3 1    19    380
ASMMTFIL.                NO   00.000  00.000  PERM  REL  NO     256    256    0    0  0         60   0   1     1    1  19910     3     60
$SYSLODLIB.              NO   00.000  00.000  PERM  SEQ  NO     256    256    0    0  0       6381   1   1     1    1   10 0   400   8000
HCR-LIB2.                NO   00.000  00.000  PERM  SEQ  NO     256    256    0    0  0        393   0   1     1    1  123 6    50   1000
IND1A.                   NO   00.000  00.000  PERM  SEQ  NO     520    100    0    0  0          0   0   1     1    1  19918     2     22
IND2A.                   NO   00.000  00.000  PERM  IND  NO     520    100    0    0  0          0   0   1     1    1  123 2     2     22
IND3A.                   NO   00.000  00.000  PERM  IND  NO     520    100    0    0  0          0   0   1     1    1  123 4     2     22
IND4A.                   NO   00.000  00.000  PERM  IND  NO     520    100    0    0  0          0   0   1     1    1  11817     2     22
$OSRSDNTLIB.             NO   00.000  00.000  PERM  SEQ  YES    256    256    0    0  0       1030   1   1     1    1   30 0   200   4000
PRF1A.                   NO   00.000  00.000  PERM  SEQ  NO      50     50    0    0  1 L        0   1   1     1    1  119 0     2     96
PRF2A.                   NO   00.000  00.000  PERM  SEQ  NO      50     50    0    0  1 L        0   1   1     1    1  120 0     2     96
PRF3A.                   NO   00.000  00.000  PERM  SEQ  NO      50     50    0    0  1 L        0   1   1     1    1   43 0     2     96
RPGERRMSG.               NO   00.000  00.000  PERM  REL  NO      84     80    0    0  0        980   0   1     1    1   91 0    49   1911
$SGOBJ.                  NO   00.000  00.000  PERM  SEQ  NO     256    256    0    0  0       4026   0   1     1    1   96 9   400   8000
SYSQJE.                  YES  00.000  00.000  PERM  SEQ  YES     18     18    0    0  0         62   1   1     1    1    3 0     1     62
$SYSPROCLIB.             NO   00.000  00.000  PERM  SEQ  NO      84     80    0    0  0        380   0   1     1    1   44 0    40   1550
IND2A.                   NO   00.000  00.000  PERM  INF  NO      24      0    2    0  0          0   0   1     1    1  118 9     4    232
$MSGLIBINPUT.            NO   00.000  00.000  PERM  SEQ  NO     256    256    0    0  0        552   0   1     1    1  116 9    40    800
IND3A*.                  NO   00.000  00.000  PERM  INF  NO      24      0    2    0  0          0   0   1     1    1  11813     4    232
IND4A*.                  NO   00.000  00.000  PERM  INF  NO      24      0    2    0  0          0   0   1     1    1  119 2     4    232
CORMTFIL.                NO   00.000  00.000  PERM  REL  NO     256    256    0    0  0        100   0   1     1    1  19913     5    100
$NUCLIB.                 NO   00.000  00.000  PERM  SEQ  NO     256    256    0    0  0          0   1   1     1    1    4 0    60   1200
$MSGLIB.                 NO   00.000  00.000  PERM  SEQ  NO     256    256    0    0  0          0   1   1     1    1    7 0    60   1200
$SYSELOG.                NO   00.000  00.000  PERM  SEQ  NO      40     40    0    0  0        520   1   1     1    1  199 0    10    520
FLRP1B.                  NO   00.000  00.000  PERM  SEQ  NO      50     50    0    0  1 L        0   1   1     1    1   41 0     2     96
LFRP1B.                  NO   00.000  00.000  PERM  SEQ  NO      50     50    0    0  1 U        0   1   1     1    1   41 0     2     96
PRFL1B.                  NO   00.000  00.000  PERM  SEQ  NO      50     50    0    0  1 U        0   1   1     1    1  119 0     2     96
$SYSMACLIB.              NO   00.000  00.000  PERM  SEQ  NO      84     80    0    0  0      19940   0   1     1    1   56 0   700  27300
PRFL2B.                  NO   00.000  00.000  PERM  SEQ  NO      50     50    0    0  1 U        0   1   1     1    1  120 0     2     96
PRFL3B.                  NO   00.000  00.000  PERM  SEQ  NO      50     50    0    0  1 U        0   1   1     1    1   43 0     2     96
THERE ARE    36 FILES IN THIS CATALOG.

VOLUME DAVES    ***SPACE MAP***

STARTING CCCHH = 11819 12516 41 2 119 6 43 2 40 0 120 2
NUMBER OF TRACKS =  1  1464   38   14   18  20    60

TOTAL NUMBER OF AVAILABLE TRACKS THIS PACK = 1615.
```

Figure 2-4. Pack Catalog Display

```
*** MEMOREX * SYSTEM * CATALOG * DISPLAY ***   DATE. . . APR 13, 1973   TIME. . .00:31:44.      PAGE. . .  2

VOLUME DAVES  *** CENTRAL CATALOG ***

FILE NAME AND QUALIFIER   MSC  CRDATE   EXDATE   STAT  TYP  VER  LBLOK  LRECD  LKY  RKP  PF&I  OFFSET    HIBLOK  ISEGS    TVOLS  VOLUME

ASMMTFIL.                 NO   00.000   00.000   PERM  REL  NO   256    256    0    0    0     00000000     60   1        1      DAVES
$OSRSDNTLIB.              NO   00.000   00.000   PERM  SEG  YES  256    256    0    0    0     00000000   1030   1        1      DAVES
$SYSPROCLIB.              NO   00.000   00.000   PERM  SEQ  NO    84     80    0    0    0     00000000    380   1        1      DAVES
$NUCLIB.                  NO   00.000   00.000   PERM  SEQ  NO   256    256    0    0    0     00000000      0   1        1      DAVES
$MSGLIB.                  NO   00.000   00.000   PERM  SEQ  NO   256    256    0    0    0     00000000      0   1        1      DAVES
FLRP1B.                   NO   00.000   00.000   PERM  SEQ  NO    50     50    0    0    1 L   00000000      0   1        1      DAVES
LFRP1B.                   NO   00.000   00.000   PERM  SEQ  NO    50     50    0    0    1 U   00000000      0   1        1      DAVES
PRFL1B.                   NO   00.000   00.000   PERM  SEQ  NO    50     50    0    0    1 U   00000000      0   1        1      DAVES
PRFL2B.                   NO   00.000   00.000   PERM  SEQ  NO    50     50    0    0    1 U   00000000      0   1        1      DAVES
PRFL3B.                   NO   00.000   00.000   PERM  SEQ  NO    50     50    0    0    1 U   00000000      0   1        1      DAVES
$CCAT                     NO   00.000   00.000   PERM  GEN  NO   128    128    0    0    0     00000000   3100   1        1
RPGERRMSG.                NO   00.000   00.000   PERM  REL  NO    84     80    0    0    0     00000000    980   1        1      DAVES
$MSGLIBINPUT.             NO   00.000   00.000   PERM  SEQ  NO   256    256    0    0    0     00000000    552   1        1      DAVES
$SYSMACLIB.               NO   00.000   00.000   PERM  SEQ  NO    84     80    0    0    0     00000000  19940   1        1      DAVES
$SRTOBJLIB.               NO   00.000   00.000   PERM  SEQ  NO   256    256    0    0    0     00000000    863   1        1      DAVES
$SYSOBJLIB.               NO   00.000   00.000   PERM  SEQ  NO   256    256    0    0    0     00000000   1869   1        1      DAVES
$SYSLODLIB.               NO   00.000   00.000   PERM  SEQ  NO   256    256    0    0    0     00000000   6381   1        1      DAVES
HCR-LIB2.                 NO   00.000   00.000   PERM  SEQ  NO   256    256    0    0    0     00000000    393   1        1      DAVES
IND1A.                    NO   00.000   00.000   PERM  SEQ  NO   520    100    0    0    0     00000000      0   1        1      DAVES
IND2A.                    NO   00.000   00.000   PERM  IND  NO   520    100    0    0    0     00000000      0   1        1      DAVES
IND3A.                    NO   00.000   00.000   PERM  IND  NO   520    100    0    0    0     00000000      0   1        1      DAVES
IND4A.                    NO   00.000   00.000   PERM  IND  NO   520    100    0    0    0     00000000      0   1        1      DAVES
PRF1A.                    NO   00.000   00.000   PERM  SEQ  NO    50     50    0    0    1 L   00000000      0   1        1      DAVES
PRF2A.                    NO   00.000   00.000   PERM  SEQ  NO    50     50    0    0    1 L   00000000      0   1        1      DAVES
PRF3A.                    NO   00.000   00.000   PERM  SEQ  NO    50     50    0    0    1 L   00000000      0   1        1      DAVES
$SGOBJ.                   NO   00.000   00.000   PERM  SEQ  NO   256    256    0    0    0     00000000   4025   1        1      DAVES
$SYSELOG.                 NO   00.000   00.000   PERM  SEQ  NO    40     40    0    0    0     00000000    520   1        1      DAVES
IND2A*.                   NO   00.000   00.000   PERM  INF  NO    24      0    2    0    0     00000000      0   1        1      DAVES
IND3A*.                   NO   00.000   00.000   PERM  INF  NO    24      0    2    0    0     00000000      0   1        1      DAVES
IND4A*.                   NO   00.000   00.000   PERM  INF  NO    24      0    2    0    0     00000000      0   1        1      DAVES
COBMTFIL.                 NO   00.000   00.000   PERM  REL  NO   256    256    0    0    0     00000000    100   1        1      DAVES
THERE ARE 31 FILES IN THIS CATALOG.

I UTSS4000 END OF CATALOG DISPLAY
```

Figure 2-5. Central Catalog Display

REBUILD TRACK – UTSSRT

The Rebuild Track utility program runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTSSRT PROGRAM CAPABILITIES

The Rebuild Track utility program prints a defective track on the printer (hexadecimally with translation). The operator then has the option of correcting data on the bad track and/or copying the bad track data to an alternate track if the user has assigned one. When the user requests alternate track assignment, corrections are made to the alternate after the track is copied. Changing the length of an EOF record is an exception to this procedure. The length of the EOF record is changed during the copy to alternate track, and the contents of the EOF record are corrected after the track is copied. The corrected data is printed for operator verification. If the user supplies neither correction data nor an alternate track assignment, UTSSRT terminates normally after the bad track display.

UTSSRT PARAMETERS

The following parameters are supplied to the Rebuild Track utility program in //PAR statements.

Keyword	Parameter	Description
TRACK=	(ccc,hh)	Decimal specification of the cylinder and head numbers of the track to be displayed. Both parameters must be supplied. Required operand.
ALTRAK=	{YES NO}	YES specifies that an alternate track will be assigned for a defective track. NO specifies that an alternate track will not be assigned for a defective track. Default is NO.

The following two parameters are required to correct data. These parameters can correct an entire record, or any number of contiguous bytes within the record. The user must supply corrective data only for the bytes he wants to change.

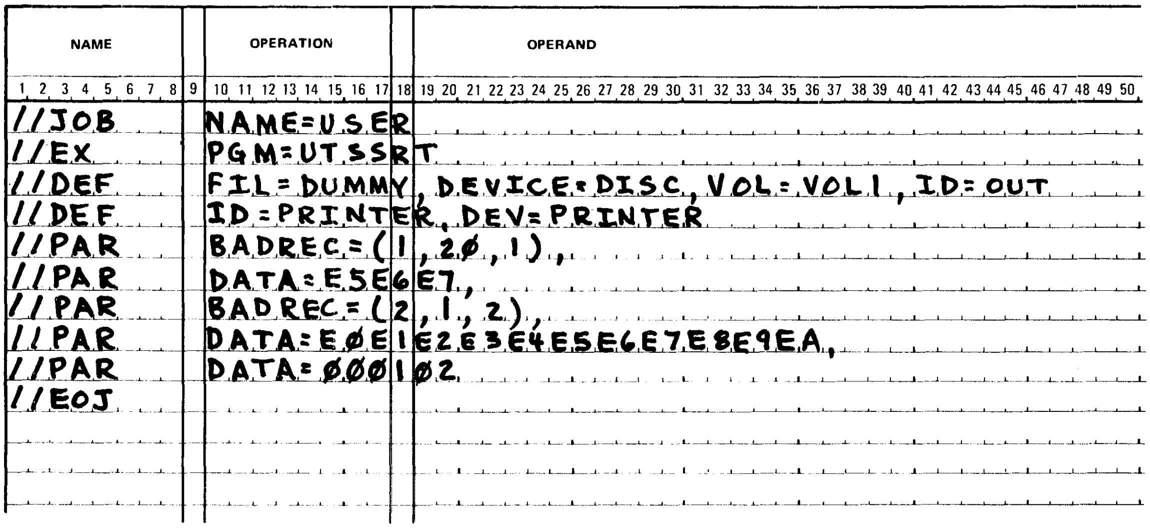
Keyword	Parameter	Description
BADREC=	(rrrr,pppp,nnn)	rrrr – required decimal specification of record number (must be 0 or more). Successive BADREC cards must have either the same record number or be in ascending order by record number. pppp – byte position within the record (must be 0 or more). nnn – number of corrective DATA cards containing data for this record (must be 1 or more). All three subparameters are required if BADREC is specified.
DATA=	n. . .n	From 2-60 hexadecimal characters/card to replace the data at byte position in the record specified by the BADREC parameter. A BADREC parameter must precede every set of DATA parameters. An even number of hexadecimal characters must be on each card; two card characters are entered in the record as a single byte.

UTSSRT REQUIRED CONTROL LANGUAGE

The UTSSRT utility program requires the standard //JOB and //EX statements and a //DEF statement with ID=OUT, FIL=DUMMY and VOL=volid for the disc volume containing the defective track. A //DEF statement with ID=PRINTER for printer output is required. //PAR statements with the utility parameters must also be included.

To correct more than one record, specify the first record location, then the data to be placed in this record. Then specify record location and data for every additional record to be corrected. The record numbers in successive BADREC cards must either be equal or in ascending order.

UTSSRT EXAMPLE



```
NAME      OPERATION  OPERAND
//JOB     NAME=USER
//EX      PGM=UTSSRT
//DEF     FIL=DUMMY,DEVICE=DISC,VOL=VOL1,ID=OUT
//DEF     ID=PRINTER,DEV=PRINTER
//PAR     BADREC=(1,2Ø,1),
//PAR     DATA=E5E6E7,
//PAR     BADREC=(2,1,2),
//PAR     DATA=EØE1E2E3E4E5E6E7E8E9EA,
//PAR     DATA=ØØØ1Ø2
//EOJ
```

The string of corrective data supplied by the DATA parameters will be entered contiguously in record 2:

E0E1. . . EA000102

STAND-ALONE MEMORY DUMP – UTSSMD

The Stand-Alone Memory Dump operates independently of MRX/OS. The operator controls program execution through the operator's console (see *Memory Dump Operator Procedure* in Appendix D for complete information on executing this utility program). UTSSMD is supplied to the user as a card deck. There are no parameters for this program. The operator loads the deck into control storage starting at address zero.

UTSSMD PROGRAM CAPABILITIES

The Stand-Alone Memory Dump utility program runs in control memory and provides an operator controlled dump of main memory (both system and user areas) and all registers except the Group III Extended registers (10-1F_{16}).

UTSSMD PROGRAM STOPS

If any of the UTSSMD program stops occurs, the dump may be retried. In some cases register contents may have been altered. The stop number appears in extended register 1 (ignoring register bits 0 and 1) of the processor state that the dump routine is being run in.

Stop Number	Meaning of Stop
11	Bounds error (should not happen)
1D	Main storage parity error*
29	Control store parity error
69	Executing processor not 1 or 2
114	Autoload attempted by user
3A7,3A8	Dump complete

The following stops should not occur. The user may retry the dump after ascertaining that the executing processor is not in stop mode, the processor selected is correct and the printer is on-line and ready.

Stop Number	Meaning of Stop
427	No response to select out on channel (including channel not there on processor 1 or 2).
42D	Channel control check
434	Channel transmission check
438	Status in fails to drop as response to select out drop in control unit busy sequence.

*A main storage parity on an exact 8K boundary of memory is interpreted as the end of a less than 65K system dump, and the dump completion stop will occur.

Stop Number	Meaning of Stop
444	Select in fails to drop as response to select out drop in device off-line sequence.
451	Neither printer 0E nor printer 1E are on-line and connected.
463	Address in fails to raise as a response to drop of address out.
478	Address of responding device not address of selected device.
484	Address in fails to drop as a response to raise of command out.
492	Status in fails to raise as a response to drop of command out.
4AD	Status in fails to drop as a response to raise of service out.
4C9	Request in fails to raise for presentation of device end status.
4DA	Service in or status in fails to raise as a response to drop of service out.*
4E5	Service in fails to drop as a response to raise of service out.
4F1	Service in fails to drop as a response to raise of command out.

DUMP OUTPUT FORMAT

The Stand-Alone Dump utility program displays:

- The 64 registers addressable by processor 0, with the exception of the Group III Extended registers 10-1F$_{16}$. The Group III registers are printed as all zeros.

- The 64 registers addressable by processors 1-7 in processor number sequence; with the exception of the Group III Extended registers 10-1F$_{16}$. The Group III registers are printed as all zeros.

*The printer is not available. If the user has two printers, either the control unit for 0E (which dominates) is on and printer 0E is not ready, or the control unit for 0E is off but printer 1E is not ready.

- The registers are printed left to right, the first line being 0-F, the next line 10-1F, etc.

- Main storage, in hexadecimal, 16 words or 32 bytes/line, with the first byte address of the line also printed. The same 16 words are also displayed in EBCDIC character form to the right of the hexadecimal format. A period is substituted for an untranslatable character. If a print line would contain all zeros, it is not printed.

See Figure 2-6, Sample Memory Dump Output.

The contents of these registers are destroyed before the registers are dumped:

	Register Number	Register Name	Processor
	19-1F	T15-T21	Line Printer
Extended Registers	1	P Micro	Line Printer
	8	Boundary Crossing	Common Block
	9	Control Storage Scan	Common Block
	A	Address Register	Common Block
	C	Address Register	Common Block

NORMAL TERMINATION

The memory dump terminates when a memory address wrap occurs on a 65K memory system or when a parity error occurs at an address in storage that is a multiple of 8K (8192_{10}).

NOTE

After Stand-Alone Memory Dump termination, MRX/OS micro-code must be reloaded.

```
PROCESSOR NUMBER 0

  BASIC    0000   15EE   0000   0000   13E2   13E4   0000   1598 .  0000   1404   1406   1408   13ED   13E2   00BA   0000
           0080   08FB   0001   0000   13E6   0000   FFE9   FFE8 .  0006   FFE6   FFE5   FFE4   FFE3   FFE2   FFE1   FFE0
  EXTND    F300   0D0A   2008   33E7   0006   0110   F080   00F8 .  0108   FF80   0141   0000   0000   0000   0000   0000
           0000   0000   0000   0000   0000   0000   0000   0000 .  0000   0000   0000   0000   0000   0000   0000   0000

PROCESSOR NUMBER 7

  BASIC    4D7E   0000   0000   6300   0000   0050   5808   5DBA .  0000   5812   580A   580C   0040   0000   00E0   0044
           0001   08FB   0000   30F8   0000   FF0A   FF09   30F7 .  FF07   FF06   FF05   FF04   FF03   FF02   FF01   FF00
  EXTND    1302   0000   2008   3863   0006   0110   F080   00F8 .  01E8   FF80   0141   0000   0000   0000   0000   0000
           0000   0000   0000   0000   0000   0000   0000   0000 .  0000   0000   0000   0000   0000   0000   0000   0000

****** END OF REGISTER DUMP ******

  0000     BA0F BA00 BA00 BA00 BA00 BA00 BA00 BA00 BA00 BA00 BA00 3033 2D31 352D 3733 8001 ................................
  0020     00F8 FE81 0107 3031 EA00 0CE8 149A 1494 E601 0028 FA01 122E FA01 0AF4 5A00 0016 .8..................Y....W.........4..
  0040     1C3C 0008 3001 EA07 162A EA07 1810 BA03 8C0F 0AF5 B001 0AF4 3519 EA07 0F28 34E1 ..........................5....4.....
  0060     F801 0A58 EA07 0776 2300 2001 FB00 1F86 B000 00C7 B001 0013 FB00 1FBA EA07 180A 8..........................G..........

  3C60     0000 8000 8000 0200 1306 0000 8000 0000 0200 130E 0000 8000 8000 0200 1300 0000 ................................
  3C80     8000 0000 0200 13DC 0000 8000 8000 0200 130B 0000 8000 0000 0200 130A 0000 8000 ................................
  3CA0     8000 0200 1309 0000 8000 0000 0200 1308 0000 8000 8000 0200 1307 0000 8000 0000 ................................
  3CC0     0200 1306 0000 8000 8000 0200 1305 0000 8000 0000 0200 1304 0000 8000 8000 0200 ................................
  3CE0     1303 0000 8000 0000 0200 1302 0000 8000 8000 0200 1301 0000 8000 0002 0013 001C ................................
  3D00     7600 0000 0200 1300 0000 0800 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 ................................

****** END OF MEMORY DUMP ******
```

Figure 2-6. Sample Memory Dump Output

STAND-ALONE DISC-TO-DISC COPY – UTSSDD

The Stand-Alone Disc-to-Disc Copy program does not run under control of MRX/OS; it executes as a stand-alone program. The user communicates with the program through the operator's console (see *UTSSDD Parmaeters* in following text). Procedures for running the program can be found under *Stand-Alone Disc-to-Disc Copy Operator Procedure* in Appendix D.

UTSSDD PROGRAM CAPABILITIES

The UTSSDD program copies a disc pack. Both the sending (original) and the receiving (copy) disc packs must have a home address and track descriptor record (record zero) written on all tracks. Each pack must have a device label, and the region designated for alternate tracks must be the same on both packs. A Memorex copy pack should have an alternate track assignment table at cylinder 0, head 0, record 4. These receiving pack format requirements can be accomplished by using the Disc Initialize utility program.

COPY PROCEDURE

This utility copies both Memorex and non-Memorex disc packs.

Non-Memorex Packs

All tracks within the operator specified limits are copied starting with record zero. If cylinder 0, head 0 falls within the specified copy limits, it is copied from the input and the contents of the output track are not saved.

A non-Memorex EOF record is assumed to have a 4-byte data length.

Memorex Packs

If cylinder 0, head 0 is included in the operator specified copy limits, the utility saves cylinder 0, head 0, records 3 and 4. After input records 1 and 2 are copied on the output disc, records 3 and 4 are rewritten on the output pack.

If the Memorex pack is non-system resident, the remaining specified tracks are copied from the input to the output pack.

If the Memorex pack is system resident, control storage data resides on a contiguous set of tracks starting at cylinder 0, head 1. This data is copied from the input disc if it appears there. If no such data is present, normal copy procedures are used.

A Memorex EOF record is assumed to have the data length specified by the first 2 bytes of the record.

UTSSDD PARAMETERS

The program requests the following parameters from the operator by printing the keywords on the operator's console. The operator enters the appropriate parameter for the keyword followed by one of the termination replies for this utility program (see text following parameters).

Keyword	Parameter	Description
FROM DRIVE=	n	A 1-digit drive number (0-7) of the drive containing the pack to be copied from. This pack must be mounted for READ ONLY. Default value is 0.
FROM VOLUME=	volid	The 1-6 character volume identifier of the disc pack to be copied. Required parameter.
TO DRIVE=	n	The 1-digit drive number (0-7) of the drive containing the pack to be copied to. This pack must be mounted for READ/WRITE. Default value is 1.
TO VOLUME=	volid	The 1-6 character volume identifier of the receiving disc pack. Required parameter.
LOWER LIMIT=	ccc,hh	Enter the lowest address of the tracks to be copied.* ccc is the 1-3 digit cylinder number (0-202) hh is the 1-2 digit head number (0-19) Default is 0, 1. If not defaulted, both parameters must be supplied.
UPPER LIMIT=	ccc,hh	Enter the highest address of the tracks to be copied.* ccc is the cylinder number (0-202) hh is the head number (0-19) Default is 199,19.

*Do not include the alternate track area (allocated at disc pack initialization) in the lower and upper copy limits area.

Keyword	Parameter	Description
MRX PACK?	YES NO	If a Memorex pack is to be copied, reply Y or YES. If a non-Memorex pack is to be copied, reply N or NO. Default value is YES.
SYSRES?	YES NO	This message is sent only if the reply to MRX PACK? was Y or YES. If a system-resident Memorex pack is to be copied, reply Y or YES. If a non-system-resident Memorex pack is to be copied, reply N or NO. Default value is NO.

TERMINATION REPLIES

Reply	Meaning	
CR	Carriage Return	
ETX	End of Text	
EOT	End of Transmission	
ETB	End of Block	
CAN	Cancel	Cancel Reply
NAK	Negative Acknowledgment	
ENQ	Inquiry	

TAPE INITIALIZE PROGRAM – UTSSTI

The Tape Initialize utility program runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTSSTI PROGRAM CAPABILITIES

The Tape Initialize utility program initializes one tape or a series of tapes. Tape initialization includes providing:

1. A standard tape volume label (see Appendix A)

2. A pair of zero-filled header labels (header 1 and header 2)

3. A tape mark

After initialization, the tape is rewound and unloaded. A console message lists the tape device address and the volume identifier of the tape just initialized. Another tape can be mounted on the drive.

UTSSTI PARAMETER

The following UTSSTI parameter is specified in a //PAR statement.

Keyword	Parameter	Description
LABEL=	{ (volid,nnn,ownerid) volid }	volid is 1-6 character identifier to be inserted as the volume serial number of the initialized tape. If volid < 6 characters, it is left-justified and blank-filled. If nnn > 1, volid must be 6 non-negative numeric characters. nnn is 1-3 digit specification of the number of tape drives to be initialized with consecutive serial numbers. If nnn > 1, the volume serial number (volid) will be incremented by one after each tape initialization and inserted as the volume serial number of the next tape initialized. (ownerid is unchanged.) The highest volume serial number assigned is 999999. The default is 1.

(continued)

Keyword	Parameter	Description
		ownerid is a 1-10 character owner identifier and name and address code to be inserted in the tape volume label of the initialized tape. If ownerid < 10 characters, it is left-justified and blank-filled. The default is blanks. Required parameter.

UTSSTI REQUIRED CONTROL LANGUAGE

The UTSSTI utility program requires the standard //JOB and //EXEC statements, a //DEF statement for each tape drive unit to be used and a //PAR statement for each set of consecutive volume serial numbers to be assigned. //DEF statement required parameters are:

$$\text{ID=OUTx,DEV=TAPE}\left\{\begin{array}{c}8\\16\\\text{cuu}\end{array}\right\}\text{,FIL=PIO,VOL=WORK}$$

where:

x   is consecutive numbers beginning at 1 and having a maximum of 4; a maximum of 4 tape drives can be used for tape initialization.

cuu   is a tape device address.

UTSSTI EXAMPLES

EXAMPLE 1

To assign the volume identifier MYTAPE and a blank owner information field to an 800 bpi tape:

NAME		OPERATION		OPERAND
1 2 3 4 5 6 7 8	9	10 11 12 13 14 15 16 17	18	19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB		NAME=XAM	P	L1
//EX		PGM=UTSS	T	I
//PAR		LABEL=MY	T	APE
//DEF		ID=OUT1,	D	EV=TAPE8, FIL=PIO, VOL=WORK
//EOJ				

EXAMPLE 2

To assign volume identifiers to the 1600 bpi tapes PAY and TEST. PAY is to have PAYROLLDEP assigned as owner identification and TEST is to be assigned no owner identification. PAY and TEST are initialized on the same tape drive; PAY is initialized first.

```
NAME     OPERATION  OPERAND
12345678 9 10-17 18 19-50
//JOB     NAME=XAMPL2
//EX      PGM=UTSSTI
//PAR     LABEL=(PAY,,PAYROLLDEP),
//PAR     LABEL=TEST
//DEF     ID=OUT1,DEV=TAPE16,FIL=PIO,VOL=WORK
//EOJ
```

EXAMPLE 3

To assign consecutive volume serial numbers (100000-100049) to 50 1600 bpi tapes. Owner information is blank. Two tape drives are used: tapes 100000, 100002,... 100048 are initialized on the drive assigned by ID=OUT1 and tapes 100001, 100003,... 100049 are initialized on the drive assigned by ID=OUT2.

```
NAME     OPERATION  OPERAND
12345678 9 10-17 18 19-50
//JOB     NAME=XAMPL3
//EX      PGM=UTSSTI
//PAR     LABEL=(1ØØØØØ,5Ø)
//DEF     ID=OUT1,DEV=TAPE16,FIL=PIO,VOL=WORK
//DEF     ID=OUT2,DEV=TAPE16,FIL=PIO,VOL=WORK
//EOJ
```

STAND-ALONE DISC-TO-TAPE DUMP – UTSSDT

The Stand-Alone Disc-to-Tape Dump utility program does not run under MRX/OS; it executes as a stand-alone program. The user communicates with the program through the operator's console (see *UTSSDT Parameters* in following text). Procedures for loading and running the program are discussed under *Stand-Alone Disc-to-Tape Dump Operator Procedures* in Appendix D.

UTSSDT PROGRAM CAPABILITIES

This utility program dumps a partial or entire disc pack on a tape file. Multiple discs can be dumped during a single execution of UTSSDT.

UTSSDT dumps any MRX disc tracks. All the records on the track are dumped with the exception of track 0. Only records 0-3 of track 0 are dumped. Record 4, the Alternate TRACK Assignment Table, is not dumped. If a primary track is defective, the assigned alternate track is dumped.

INPUT/OUTPUT SPECIFICATIONS

UTSSDT requires that the disc tracks to be dumped and track zero, whether or not it is to be dumped, be Memorex system output. This requirement provides that:

- Any disc device label must be a Memorex device label
- Any end of file records dumped must be Memorex end of file records
- All tracks to be dumped must contain at least a home address field (Figure 2-1) and a track descriptor record (Figure 2-2). The length of the track descriptor data record (except for cylinder 0 head 0) must be 8 bytes.
- A defective primary track must have an alternate assigned and specified in the track descriptor record count field. The home address flag byte must show that the track is defective.

UTSSDT output tapes must contain the standard tape label information (see UTSSTI utility in this section). The output tape will be written by UTSSDT in a format conducive to restoration by the Tape-to-Disc Restore utility.

- The tape drives used must meet MRX 3237 tape subsystem specifications
- Each disc pack track is compressed into a single tape block
- A header precedes the data on tape and includes at least:

  1. Tape block number

2. Word of flags for internal use

3. Primary* track cylinder and head number

UTSSDT PARAMETERS

The program requests the following parameters from the operator by printing the keywords on the operator's console. The operator enters the appropriate parameter for the keyword followed by a carriage return or ETX. The program validates the parameter and either repeats the parameter request or requests the next parameter.

Keyword	Parameter	Description
FROM DISC (drive,volume)=	n,volid	n is the 1-byte disc drive number of the disc drive containing the disc pack to be dumped. volid is the 1-6 byte volume identifier in the disc device label. UTSSDT compares volid with the actual volume identifier. If they do not match, the actual volume identifier is printed and permission to continue is requested. If denied, the original parameter request is reissued. Otherwise, parameter requests continue. A required parameter; both subparameters are also required.
TO TAPE (cuu, volume)=	cuu, volid	cuu is the device address, 1-digit channel number and 2-digit unit number, of the tape drive on which the first output tape is mounted. volid is the 1-6 byte volume identifier in the tape volume label.

*Including the primary track cylinder and head number and omitting the Alternate Track Assignment Table aid in restoring the tape to a disc pack other than the disc pack that was dumped.

Keyword	Parameter	Description
		UTSSDT compares volid with the actual volume identifier. If they do not match, the actual volume identifier is printed and permission to continue is requested. If denied, the original parameter request is reissued. Otherwise, parameter requests continue. A required parameter; both subparameters are also required.
DUMP ENTIRE PACK?	{Y N}	Y indicates that the entire pack (from cylinder 0, head 0 through cylinder 199, head 19) is to be dumped. Parameter requests terminate and the dump begins. N indicates that only selected tracks are to be dumped. The limits parameter requests are then issued. The region allocated for alternate track assignment at initialize time must not be included within the partial dump limits. The default is Y.
LOWER LIMIT (cyl,hd)=	ccc,hh	This parameter specifies the starting address of a range of tracks to be dumped. ccc is the 1-3 digit cylinder number ranging from 0 to 202. hh is the 1-2 digit head number ranging from 0 to 19. A required parameter; also both subparameters are required.
UPPER LIMIT (cyl,hd)=	ccc,hh	This parameter specifies the ending address of the range of tracks to be dumped. ccc is the 1-3 digit cylinder number ranging from 0 to 202. hh is the 1-2 digit head number ranging from 0 to 19.

Keyword	Parameter	Description
		If the upper limit < lower limit, the LOWER LIMIT parameter request is reissued. The default is the lower limit; one track is dumped.
MORE TRACKS THIS PACK?	Y N	This parameter request is issued only after a partial dump. Y indicates that more tracks on this pack are to be dumped. The limits parameter requests are reissued. N indicates that no more tracks are to be dumped from this pack. The default is N.
DUMP ANOTHER PACK?	Y N	Y indicates that another pack is to be dumped before the program terminates. The utility program begins reissuing parameter requests starting with the FROM DISC parameter. N indicates that pack dumping is finished. The default is N.

OPERATOR INFORMATION

The Disc-to-Tape Dump utility program operates independent of MRX/OS. The operator controls program execution through the operator's console (see Appendix D for detailed procedures). When the program is loaded it prints a header message and begins printing a series of parameter requests. After each request, the operator enters the appropriate parameter and a carriage return or ETX. Other normal termination replies for this utility program are:

EOT	End of Transmission	
ETB	End of Block	
CAN	Cancel	cancel reply
NAK	Negative Acknowledgment	
ENQ	Inquiry	

If, during the dump, the utility detects an end-of-tape indicator, the utility displays the message:

MORE TAPE IS NEEDED

When a new tape is ready, the FROM TAPE parameter is repeated.

The normal program completion message is

DISC-TO-TAPE DUMP COMPLETE.

Output tapes are rewound and unloaded.

UTSSDT PROGRAM STOPS

The following program stop numbers can appear in the processor 4 P-register. When the error has been corrected, the program must be restarted at address zero (step 4 of operating procedures).

Stop Number — Meaning of Stop

0002 A branch to this address has been preceded by the console message:

TAPE ERROR . . .
RETRY?

The branch takes place immediately if the user replies N to the message.

If, however, the user replies Y, a branch to this address occurs only if an attempt to backspace the tape in order to retry a write has failed.

The error may be caused by a defective tape or a defective tape drive. Correct the error and restart the utility.

0004 A branch to this address means that a modem or disconnect error has occurred for the operator's console. Correct the error and restart the utility.

0006 A branch to this address occurs when the user replies N to the message:

$\left\{\begin{matrix}\text{TAPE}\\ \text{DISC}\end{matrix}\right\}$ UNSOLICITED ATTENTION
CONTINUE?

Stop Number	Meaning of Stop
0008	A branch to this address is preceded by the message: DEVICE cuu TIMED OUT During the dump this indicates that there is a hardware malfunction; specifically, channel-end is not being returned within a specified amount of time. When this condition occurs, the utility does not send any other message before branching to this address. During parameter solicitation, however, the utility branch to this address occurs after the timeout error message only if the user has replied N to a subsequent RETRY? . The same class of hardware malfunction is implied. Recommended action: Take a memory dump and forward it with the console log to your Field Service Representative.
000A thru 0012	Stops at these addresses, should not occur. Recommended action: Dump memory and forward the dump and the console log to the Field Service Representative.
0014	A branch to this address is caused by an irrecoverable disc search failure. This indicates that either the disc is defective or the drive is malfunctioning.

STAND-ALONE TAPE-TO-DISC RESTORE – UTSSTD

The Stand-Alone Tape-to-Disc Restore utility program does not run under MRX/OS; it executes as a stand-alone program. The user communicates with the program through the operator's console (see *UTSSTD Parameters* in the following text). Procedures for loading and running the program are discussed under *Stand-Alone Disc-to-Tape Dump Operator Procedures* in Appendix D.

UTSSTD PROGRAM CAPABILITIES

This utility program restores a disc pack previously dumped on a tape by the Stand-Alone Disc-to-Tape Dump utility program. The tape drives used must meet the specifications of the MRX 3237 Subsystem. UTSSTD uses the UTSSDT output tape to restore all disc tracks dumped by UTSSDT. UTSSDT output tapes (UTSSTD input tapes) have the following characteristics:

- Each disc pack track is compressed into a single tape block (with the UTSSDT program used to read the track).

- Data on tape is preceded by a header that includes at least a tape block number, a flag word for internal use, and the cylinder and head number of the primary track from which the dump is made.

- No data from defective tracks is contained on the tapes. UTSSDT does not dump a defective track; it writes its alternate with the address (cylinder and head) of the primary track. UTSSTD does not write on defective tracks, if the primary track is defective, it writes on the assigned alternate track.

UTSSTD PARAMETERS

This program requests the following parameters from the operator by printing the keywords on the operator's console. The operator enters the appropriate keyword followed by a carriage return or an ETX.

Keyword	Parameter	Description
FROM TAPE (cuu,volume)=	cuu,volid	cuu is the device address: 1-digit channel number and 2-digit unit number of the tape drive on which the first tape to be restored to disc is mounted. volid is the 1-6 byte volume identifier in the tape volume. UTSSTD compares volid with the actual volume identifier. If they do not match, the actual volid is printed and permission to continue is requested. If denied, the parameter request is reissued. Otherwise, parameter requests continue.

Keyword	Parameter	Description
TO DISC (drive,volume)=	n,volid	n is the 1-byte drive number of the disc drive containing the disc pack to be restored. volid is the 1-6 byte volume identifier in the disc device label. UTSSDT compares volid with the actual volume identifier. If they do not match, the actual volume identifier is printed and permission to continue is requested. If denied, the parameter request is reissued. Otherwise, parameter requests continue. A required parameter; both subparameters are required.
RESTORE ANOTHER PACK?	{YES NO}	YES indicates that another disc pack is to be restored. NO indicates that restoration is finished. Default is NO.
CHANGE DISC LABEL?	{YES NO}	This parameter request is issued only if track zero (cylinder 0, head 0) is on the tape and the volume label of the disc dumped differs from the disc volume label of the disc being restored. YES or Y indicates that the tape volume label of the dumped disc should be written on the disc being restored. NO or N indicates that the previously saved volume label record of the disc being restored should be written. In all track zero restoration, the Alternate Track Assignment Table saved from the disc is written.

OPERATOR INFORMATION

The Tape-to-Disc Restore utility program operates independent of MRX/OS. The operator controls program execution through the operator's console (see Appendix D for detailed procedures). When the program is loaded, it prints a header message and begins printing a series of parameter requests. After each request, the operator enters the appropriate parameter and a carriage return or ETX. Other normal termination replies for this utility program are:

EOT	End of Transmission	
ETB	End of Block	
CAN	Cancel	cancel reply
NAK	Negative Acknowledgment	
ENQ	Inquiry	

If the utility detects an EOF on tape, the RESTORE ANOTHER PACK? message is sent. An EOV effects tape rewinding and unloading. The following message is sent to the operator:

NEXT BACKUP TAPE REQUIRED.

The FROM TAPE parameter request is then reissued to ask for the address and volume identifier of the next backup tape. When restoration is completed, the following message is sent:

TAPE-TO-DISC RESTORE COMPLETE.

UTSSTD PROGRAM STOPS

The following program stop numbers can appear in the processor 4 P-register. When the error has been corrected the program must be restarted at address zero (step 4 of operating procedures).

Stop Number	Meaning of Stop
0002	A branch to this address occurs after display of the message TAPE ERROR . . . RETRY? followed by an operator reply of N. An automatic backspace and retry of the read sequence has occurred 5 times before UTSSTD displays this message. Cause for this persistent failure may be a defective spot on the tape or a defective tape drive. If possible, correct the error and restart the utility.

Stop Number	Meaning of Stop
0004	Modem or disconnect error for the operator's console. Correct the error and restart the utility.
0006	A branch to this address occurs when the user replies N to the message: $\left\{\begin{matrix}\text{TAPE}\\\text{DISC}\end{matrix}\right\}$ UNSOLICITED ATTENTION CONTINUE? *or* to the message BYPASS TRACK? If no error has actually occurred, restart the utility.
0008	DEVICE cuu TIMED OUT is the message which precedes a branch to this address. During the actual restore, this message and program stop indicate a hardware malfunction; specifically, channel-end is not being returned within a specified amount of time. When this situation occurs, no other message appears before the program stop. During parameter solicitation, however, the message RETRY? follows the timeout message. The branch occurs only if the user has replied N. The actual timeout is a timeout of the SIO instruction. This may occur during a normal I/O attempt or during the polling sequence. Recommended action: Take a dump and forward it and the console log to your Field Service Representative.
000A thru 000E	Stops at these addresses, should not occur. Recommended action: Take a memory dump (preserving the micro-registers). Forward it and the console log to your Field Service Representative.
0010	Irrecoverable disc search failure persists after 10 retries. Either the disc is in error or the drive is malfunctioning. If the error does not clear up after restarting the utility, reinitialize the disc. If the error persists on the re-initialized pack, dump memory, and forward the dump and console log to the Field Service Representative.

LOAD UNIVERSAL CHARACTER SET UTILITY – UTSSLU

The Load Universal Character Set utility runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTSSLU PROGRAM CAPABILITIES

This program uses the LOADUCS service request (see Appendix F) to load a 240-byte image memory buffer on MRX/OS supported printers that are equipped to handle image memory load commands.

UTSSLU PARAMETER

The UTSSLU parameter is entered on a //PAR statement.

Keyword	Parameter	Description
UCS=	$\left(\begin{Bmatrix}\text{XXXX}\\ \text{CARD}\\ \text{member}\end{Bmatrix}, \begin{Bmatrix}\underline{\text{NOFOLD}}\\ \text{FOLD}\end{Bmatrix}, \text{VER}\right)$	XXXX is one of the following standard images (see Appendix F) or a load module on $NUCLIB*: AN PN SN HN QN TN PCAN QNC XN PCHN RN YN CARD indicates that the image is in card format and is located in the source file with ID=CARD. member is the 1-8 character name of a member that is to be loaded. The member specified must have been assembled and link-edited as an absolute load module in the library specified on the //DEF statement with ID=LIB. NOFOLD indicates that the image load is to be performed with no folding performed on the print data. All data byte bits are compared when searching for the graphic to fill the print position.

*A load module is specified with a standard name of UC01 through UC10. Load module names UC01-UC10 are associated with $NUCLIB names 95F1-95FA, respectively.

Keyword	Parameter	Description
		FOLD indicates that the image load is to be performed with folding; bits 0 and 1 are ignored in the compare when searching for the graphic to fill the print position. VER indicates that the image buffer is to be verified – printed for visual verification by the operator.

CHARACTER IMAGE CARD FORMAT

If UCS=CARD is specified, the file specified on the //DEF statement with ID=CARD contains image representation of character set image. CARD can be a spooled data file or defined as a preloaded source file. The CARD file contains four data cards and a variable number of comment cards. The data cards must have any of the 256 EBCDIC bit configurations punched in columns 11-70 (see Appendix F) with the exception of the NULL or BLANK. Comment cards must have asterisks in columns 1 and 2 and may have comments in columns 3-80.

UTSSLU REQUIRED CONTROL LANGUAGE

The UTSSLU utility program requires the standard //JOB and //EXEC and //EOJ statements, a //DEF statement with ID=PRT and FIL=PIO for the printer on which the load is to be performed, a //DEF statement for the CARD file if ID=CARD was specified or a //DEF statement with ID=LIB if a member of a library is to be loaded as the UCS image. The parameter is specified in a //PAR statement.

UTSSLU EXAMPLES

The following examples illustrate four methods of submitting an image to UTSSLU.

EXAMPLE 1

To define the standard EBCDIC character set as the image to be loaded. No folding or verification is performed.

NAME		OPERATION		OPERAND
1 2 3 4 5 6 7 8	9	10 11 12 13 14 15 16 17	18	19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB		NAME=LOA	D	I
//EXEC		PGM=UTSS	L	U
//PAR		UCS=AN		
//DEF		ID=PRT, D	E	V=21E, FIL=PIO
//EOJ				

EXAMPLE 2

To define the preferred character set for commercial applications of FORTRAN and COBOL to be loaded in fold mode. The operator must verify that the proper train or chain is mounted prior to loading the buffer.

```
NAME       OPERATION    OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB     NAME=LOAD2
//EXEC    PGM=UTSSLU
//PAR     UCS=(RN,FOLD,VER)
//DEF     ID=PRT,DEV=20E,FIL=PIO
//EOJ
```

EXAMPLE 3

To define a non-standard buffer image. The image is on file CARD. No folding or verification is performed.

```
NAME       OPERATION    OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB     NAME=LOAD3
//EXEC    PGM=UTSSLU
//PAR     UCS=CARD
//DEF     ID=PRT,DEV=21E,FIL=PIO
//DEF     ID=CARD,FIL=UCS1
//DATA    FIL=UCS1
       •
       •
    USC Image
       •
       •
/*
//EOJ
```

EXAMPLE 4

To define a non-standard buffer image. A //DEF card with ID=LIB is supplied since the image name is non-standard. The member UCSIMAGE will be loaded from the file UCS1. Folding is specified. Verification of print arrangement is not performed.

```
NAME      OPERATION          OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB     NAME=LOAD4
//EXEC    PGM=UTSSLU
//PAR     UCS=(UCSIMAGE,FOLD)
//DEF     ID=PRT,DEV=20E,FIL=PIO
//DEF     ID=LIB,FIL=UCS1,DEV=DISC,VOL=MRX1ØØ
//EOJ
```

3. FILE-TO-FILE UTILITY PROGRAM - UTFF

The file-to-file utility program runs under control of MRX/OS and executes in an 8K partition. The user supplies Control Language Statements to load and execute the program. In addition to the hardware requirements described in Section 1, the UTFF program also requires that two tape units be attached if tape transfer is to be performed.

UTFF PROGRAM CAPABILITIES

The File-to-File utility program, UTFF, transfers files from one medium to another. A file can be transferred from:

- Card to card, tape, disc or printer
- Tape to card, tape, disc, or printer
- Disc to card, tape, disc, or printer

This utility enables the user to:

- Transfer a file
- Create a back-up file
- Create a new file type (for example, convert a sequential file to an indexed file)

The following chart defines the specific file types that UTFF transfers.

Output Input	Card File (sequential)	Printer (sequential)	Tape File (sequential)	Disc File		
				Sequential	Indexed	Relative
Card file (sequential)	X	X	X	X	X	
Tape file (sequential)	X	X	X	X	X	
Disc file sequential	X	X	X	X	X	
indexed	X	X	X	X	X	
relative	X	X	X	X		X

INPUT AND OUTPUT CAPABILITIES AND RESTRICTIONS

The UTFF utility program handles a column binary input from the card reader and provides printer output in character and hexadecimal representation in either list or display format.

The UTFF I/O restrictions are:

- Card output: records must not be greater than 80 bytes.
- Record size: input and output files must have the same record size. If the records are variable in length, the maximum record size must be equal.
- Record length: record length must be fixed if input is not in CSDF (Common Stored Data Format) and output is in CSDF.
- Record format: if input is tape or disc in CSDF, then output tape or disc must be in CSDF.
- Block size: input and output block size must be equal when transferring relative files.
- File type: the output file type can be relative only if the input file type is relative.
- Files created by the MRX/OS Librarian cannot be copied because of internal end of files between members. Refer to the **MRX/OS Program Library Services Reference** manual for copy procedures.

Violation of the above restrictions causes an error to be displayed and causes the termination of UTFF.

UTFF PARAMETERS

The following parameters are specified in //PAR statements. The format of this statement is:

```
//PAR  option, . . .
```

PRINTER OUTPUT PARAMETERS

Keyword	Parameter	Description
SPACE=	n	The number of lines to skip between records. Maximum value of n is 3. Default value of n is 1.
PAGHDR=	'header'	An apostrophe enclosed 1-132 character heading to be printed at the top of each page. An apostrophe embedded within the heading must be represented as two adjacent apostrophes.

Keyword	Parameter	Description
		The header entry is the only UTFF parameter that may require a continuation card. The terminating apostrophe may be omitted from the first parameter card enabling the next card to continue the statement from the previous card; the continuation statement begins with the first non-blank character after the Parameter command.
LIST=	C X T	The LIST keyword effects printer output in a list format; each record begins on a new line. Not allowed if DISPLAY parameter used. If neither LIST nor DISPLAY is used, default is LIST=C. C Character output – data must be in EBCDIC format. X Hexadecimal output – 2 hexadecimal characters printed for each byte. T Hexadecimal output – 2 hexadecimal characters printed for each byte and a line of translated characters is printed above (untranslatable characters are represented by a period.
DISPLAY=	C X T	The DISPLAY keyword causes printer output in a display format. The first sixteen print positions describe the file; block number, block size, record number. Input must be disc or tape. Not allowed if LIST parameter used. If neither LIST nor DISPLAY is used, default is LIST=C. C Character output X Hexadecimal output T Hexadecimal output with a translation line printed above.

CARD INPUT PARAMETERS

COLBIN=	Y N	Y Specifies that card input is in column binary code.

Keyword	Parameter	Description
		N Specifies that card input is not in column binary code. Default is N. This option is ignored if the input device is not the card reader.
TAPE I/O PARAMETERS		
BLKSIZ=	(n,n)	The first entry refers to tape input block size and the second entry refers to tape output block size. Default is record size (+CSDF if data is in CSDF). The n entry must be less than 32,767 bytes. The comma must be included if only the output block size is present.
RECSIZ=	n	The record size, which must be less than 32,767 bytes. Default is 80 bytes.
DISC I/O PARAMETERS		
RKP=	n	Position of the key within the input record on which the index is to be built. Range of n is 0 through record size – 1. The RKP option is ignored unless the input is sequential and the output is indexed. The key remains in the record. Default is 0 bytes.
TAPE OR DISC I/O		
CSDF=	$\begin{Bmatrix}Y, Y\\ N, N\end{Bmatrix}$	Specifies whether the input and/or output files are in CSDF. The first position denotes the input file and the second position denotes the output file. The comma is required. The first position of the parameter is ignored if the input file is not tape or disc, and the second position is ignored if the output file is not tape or disc.

Keyword	Parameter	Description
		CSDF=(Y,N) would be illegal if the input and output files are on tape and/or disc.
		Y Yes, file is in Common Stored Data Format.
		N No, file is not in Common Stored Data Format.
		Default value is Y.

GENERAL OUTPUT PARAMETERS

SPOOL=	YES	This parameter, if used, specifies that the output device is spooled on disc. It can be used for either printer or punch output. It must be coded as shown.

UTFF REQUIRED CONTROL LANGUAGE

The UTFF utility program requires the standard //JOB and //EX statements and two //DEFINE statements (one for the input file with ID=INPUT, and one for the output file with ID=OUTPUT). The Parameter statements give any other necessary I/O information. Meaningless keywords in the //PAR statements are ignored.

UTFF EXAMPLES

CARD TO CARD

To reproduce a card deck,use the following CLS:

```
NAME      OPERATION  OPERAND
//JOB     NAME=CARDCARD, USER=MRX
//EX      PGM=UTFF, TIME=2
//DEF     ID=INPUT, DEV=SYSCRD
//DEF     ID=OUTPUT, DEV=CRDPCH
//DATA    FIL=SYSCRD
        •
        •
        •
      Data cards
        •
        •
        •
/*
//EOJ
```

CARD TO DISC

To transfer a column binary coded card file to a CSDF disc file use the following CLS:

```
NAME      OPERATION         OPERAND
//JOB    NAME=CARDISC,USER=YOU
//EX     PGM=UTFF,TIME=2
//DEF    ID=INPUT,DEV=SYSCRD
//DEF    ID=OUTPUT,NUM=7ØØØ,SIZ=8Ø,BLK=(5Ø,5Ø)
//PAR    COLBIN=Y
//DATA   FIL=SYSCRD
        •
        •
     Data cards
        •
        •
        •
/*
//EOJ
```

CARD TO PRINTER

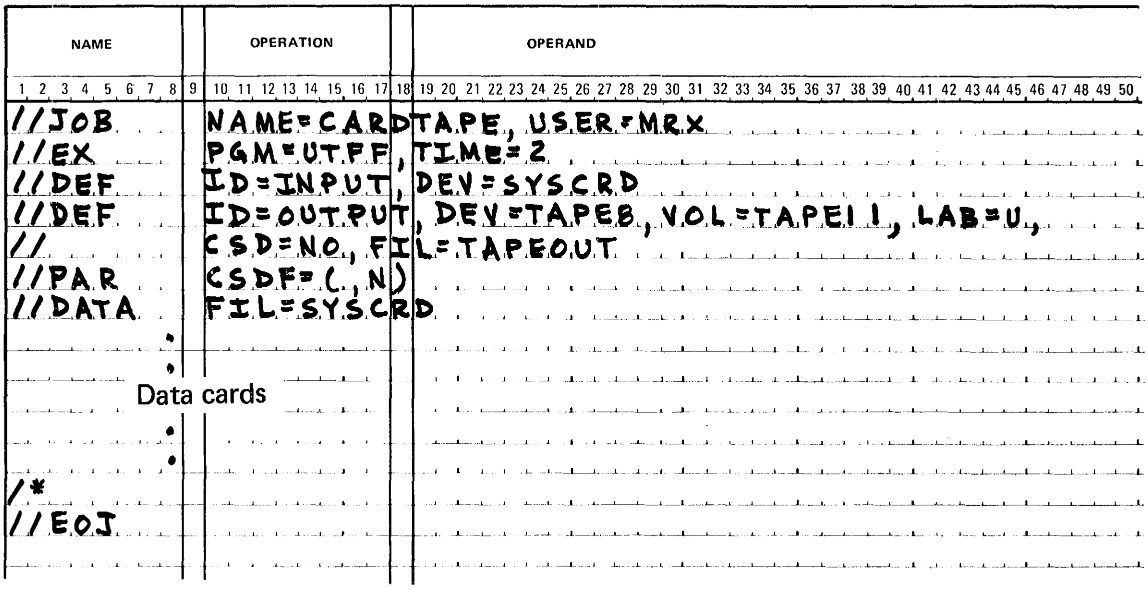
To list an EBCDIC card file in character output; the output listing is single spaced, and NAME AND ADDRESS FILE is printed at the top of each page:

```
NAME      OPERATION         OPERAND
//JOB    NAME=CARDPRNT,USER=YOU
//EX     PGM=UTFF,TIME=3
//DEF    ID=INPUT,DEV=SYSCRD
//DEF    ID=OUTPUT,DEV=PRINTER
//PAR    PAGHDR='NAME AND ADDRESS FILE'
//DATA   FIL=SYSCRD
        •
        •
     Data cards
        •
        •
/*
//EOJ
```

The SPACE and LIST options are not included on the //PAR card because the default values are acceptable.

CARD TO TAPE

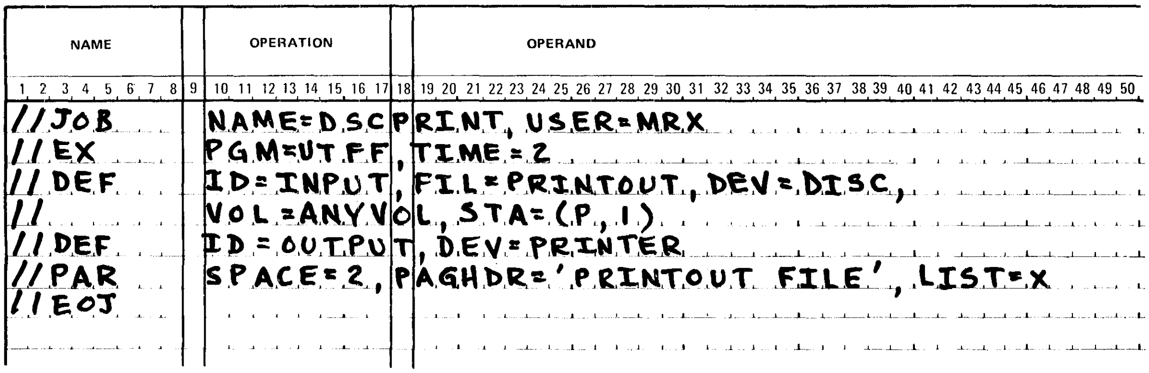
To transfer a card deck to a sequential tape file in non–CSDF, the following CLS is used:



```
NAME            OPERATION    OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB     NAME=CARDTAPE,USER=MRX
//EX      PGM=UTFF,TIME=2
//DEF     ID=INPUT,DEV=SYSCRD
//DEF     ID=OUTPUT,DEV=TAPE8,VOL=TAPE11,LAB=U,
//        CSD=NO,FIL=TAPEOUT
//PAR     CSDF=(,N)
//DATA    FIL=SYSCRD
        .
        .
     Data cards
        .
        .
/*
//EOJ
```

DISC TO PRINTER

To print each record of a CSDF disc file, PRINTOUT, in hexadecimal with double spacing between each line and the heading PRINTOUT FILE printed at the top of each page use the following CLS:



```
NAME            OPERATION    OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB     NAME=DSCPRINT,USER=MRX
//EX      PGM=UTFF,TIME=2
//DEF     ID=INPUT,FIL=PRINTOUT,DEV=DISC,
//        VOL=ANYVOL,STA=(P,1)
//DEF     ID=OUTPUT,DEV=PRINTER
//PAR     SPACE=2,PAGHDR='PRINTOUT FILE',LIST=X
//EOJ
```

DISC TO TAPE

To transfer a sequential disc file to a sequential tape file that is unlabeled, use the following CLS:

```
  NAME      OPERATION     OPERAND
12345678 9 1011121314151617 18 19202122232425262728293031323334353637383940414243444546474849 50
//JOB     NAME=DISCTAPE,USER=YOU
//EX      PGM=UTFF,TIME=1
//DEF     ID=INPUT,FIL=DISC1
//DEF     ID=OUTPUT,DEV=TAPE8,VOL=UPDATE,FIL=TP1,
//        LAB=U
//PAR     BLKSIZ=(,n).,RECSIZE=n
//EOJ
```

Block size and record size should be set to the desired sizes.

DISC TO CARD

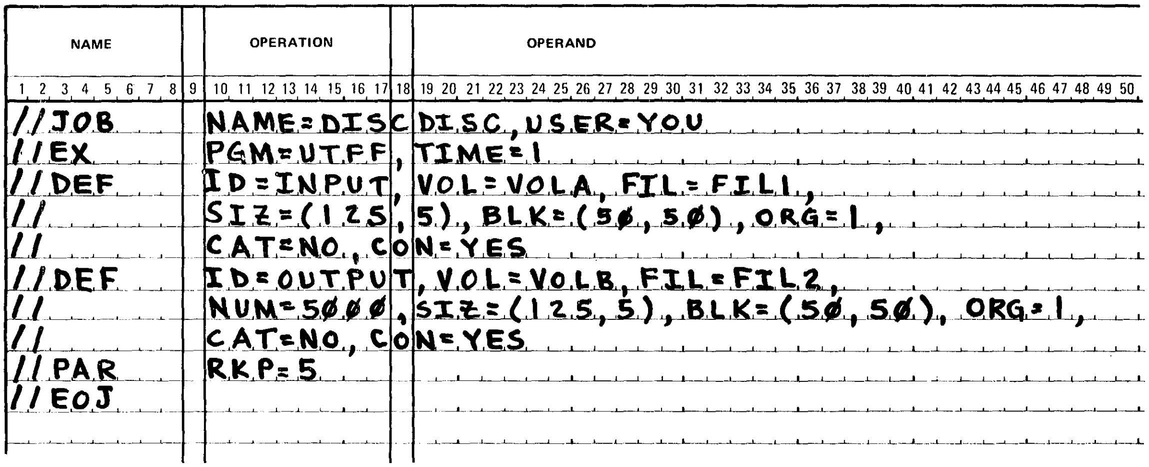
To punch a card file from the CSDF disc file, BILL, use the following CLS:

```
  NAME      OPERATION     OPERAND
12345678 9 1011121314151617 18 19202122232425262728293031323334353637383940414243444546474849 50
//JOB     NAME=DISCARD,USER=YOU
//EX      PGM=UTFF,TIME=1
//DEF     ID=INPUT,FIL=BILL
//DEF     ID=OUTPUT,DEV=CRDPCH
//EOJ
```

The //PAR card is not needed because the default value of the CSDF= option is acceptable.

DISC TO DISC

To copy a disc index sequential file, the following CLS is used:

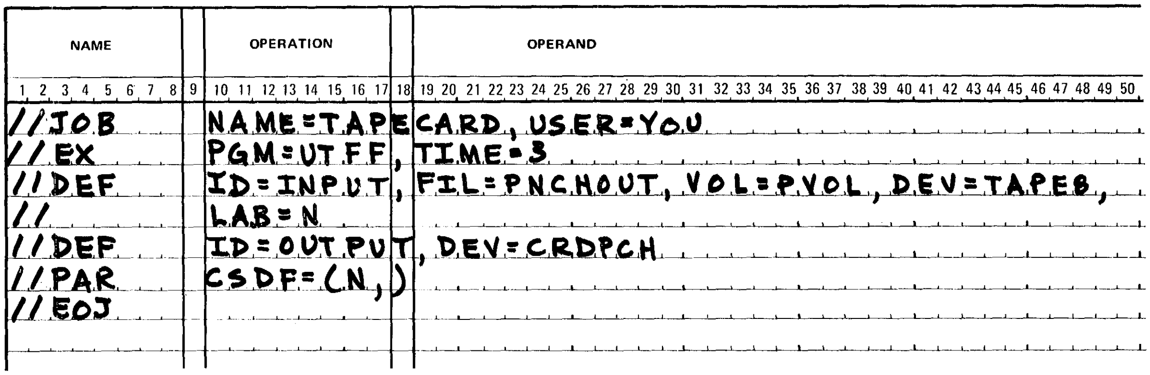


NAME
OPERATION
OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

```
//JOB    NAME=DISCDISC,USER=YOU
//EX     PGM=UTFF,TIME=1
//DEF    ID=INPUT,VOL=VOLA,FIL=FIL1,
//       SIZ=(125,5),BLK=(5Ø,5Ø),ORG=1,
//       CAT=NO,CON=YES
//DEF    ID=OUTPUT,VOL=VOLB,FIL=FIL2,
//       NUM=5ØØØ,SIZ=(125,5),BLK=(5Ø,5Ø),ORG=1,
//       CAT=NO,CON=YES
//PAR    RKP=5
//EOJ
```

TAPE TO CARD

To punch a tape file that is not in CSDF and does not have a standard label, use the following CLS:



NAME
OPERATION
OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

```
//JOB    NAME=TAPECARD,USER=YOU
//EX     PGM=UTFF,TIME=3
//DEF    ID=INPUT,FIL=PNCHOUT,VOL=PVOL,DEV=TAPE8,
//       LAB=N
//DEF    ID=OUTPUT,DEV=CRDPCH
//PAR    CSDF=(N,)
//EOJ
```

The block size for the input tape is assumed to be 80 bytes in length. Therefore, it is not included on the //PAR card as an option.

TAPE TO DISC

To transfer a sequential tape file to an indexed sequential disc file use the following CLS (tape file must first be sorted on the index sequential file key):

```
   NAME           OPERATION                  OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB    NAME=TAPEDISC, USER=MRX
//EX     PGM=UTFF,TIME=4
//DEF    ID=INPUT,DEV=TAPE8, VOL=SERIAL, FIL=TAPES,
//       LAB=S
//DEF    ID=OUTPUT, DEV=DISC, VOL=DIRACC,
//       FIL=INDXDILE, ORG=1, SIZE=(1ØØ, 1Ø),
//       NUM=1ØØØ, BLK=(2, 1Ø)
//PAR    CSDF=(,Y),RKP=8, BLKSIZ=(1ØØ)
//EOJ
```

TAPE TO PRINTER

To print an EBCDIC tape file in character form and display format with double spacing, use the following CLS:

```
   NAME           OPERATION                  OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB    NAME=TAPRNT, USER=YOU
//EX     PGM=UTFF, TIME=2
//DEF    ID=INPUT, DEV=TAPE8
//DEF    ID=OUTPUT, DEV=PRINTER
//PAR    SPACE=2, DISPLAY=C, CSDF=(Y, ), BLKSIZ=(n)
//EOJ
```

The block size should be equal to the tape block size being input.

TAPE TO TAPE

To create a tape back-up file, the following CLS is used:

```
NAME        OPERATION          OPERAND
//JOB    NAME=TAPETAPE, USER=YOU
//EX     PGM=UTFF, TIME=1
//DEF    ID=INPUT, DEV=TAPE8, FIL=MASTER
//DEF    ID=OUTPUT, DEV=TAPE8, FIL=MASTER
//PAR    BLKSIZ=(n, n), RECSIZ=n
//EOJ
```

The block size and record size should be set to the proper sizes. The input tape does not have to be blocked.

UTFF PERFORMANCE CHARACTERISTICS

Buffer space is dependent on the input/output module used and whether or not output is to the printer.

The following chart gives the maximum buffer space available for each input/output module in an 8K partition.

Output Input	Blocked I/O Non-CSDF	Logical I/O Sequential	Logical I/O Relative	Logical I/O Indexed
Blocked I/O Non-CSDF	3370* 524**			1856*
Logical I/O Sequential		3370* 524**		1856*
Logical I/O Relative		4624* 1774**		
Logical I/O Indexed		2056* 738**		3706*

*The input and output block size must not exceed this number (non-printer output).
**The input block size must not exceed this member (printer output).

NOTE

The figures listed above were calculated with a partition size of 8,200 bytes.

4. MEMOREX CONVERSION UTILITY PROGRAMS

These Memorex utility programs enable the user to convert:

- IBM disc volume to Memorex disc volume
- Converted Memorex disc volume back to IBM disc volume
- IBM tape file to Memorex tape file
- Emulated IBM M20 disc file to MRX disc file
- IBM M20 disc file to MRX tape file

The conversion utilities use the standard control language described under *Control Language* in Section 1 and can be executed as routine steps in the job stream.

IBM DISC VOLUME TO MEMOREX DISC VOLUME – UTCVIM

This program runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTCVIM PROGRAM CAPABILITIES

This disc conversion utility program readies an IBM disc file for Memorex Operating System access by generating a Memorex pack catalog from the IBM volume table of contents (VTOC). The IBM disc pack must not be mounted until UTCVIM requests that the pack be mounted.

IBM FILE REQUIREMENTS

If the converted IBM file is to be acceptable to Memorex read/write I/O, it must have the following characteristics:

- IBM sequential data set characteristics: format F records, every track except the last is filled to capacity, no record truncated except the last.
- Sequential organization
- No user header or trailer labels
- Characters 1-17 of each filename unique
- Entire file on same volume
- No split cylinder allocated

If the file to be converted does not fit the description above, UTCVIM processes it as a general file (unknown data organization). In order to access this file after conversion, the user must determine the record and file structure and use physical I/O.

PACK CATALOG GENERATION

The utility program makes a pack catalog entry for each VTOC entry, for the pack catalog, and for the VTOC. No entries are made in the central catalog. Pack catalog file names are truncated to seventeen characters. Valid file name characters are alphanumerics, $, and -. If an IBM filename contains other characters, it should be changed before UTCVIM execution.

When converting a DOS VTOC, fixed length record files must be identified by file name in the FILENMS parameter. Other file names will be entered into the pack catalog with a file type of general.

See **MRX/OS Control Program and Data Management Services Reference** manuals for pack catalog formats and descriptions.

PACK CATALOG SPACE ALLOCATION

When a file is to be used by both Memorex and IBM systems, the IBM system must allocate a file for the Memorex pack catalog prior to execution of the UTCVIM program. Additional file space may be allocated by either system after conversion. (This space would be unknown to the other system and overwritten.)

Space allocated for the pack catalog must be contiguous space, begin on a cylinder boundary, and consist of at least as many tracks as the VTOC.

OTHER CONVERSION MODIFICATIONS

The UTCVIM utility program makes modifications in addition to pack catalog generation:

- Writes an FDT for the pack catalog on cylinder 0, head 0, record 2 (cylinder 0, head 0 previously having been saved on an unused track and an entry for this track made in the pack catalog).

- Writes Memorex device label on cylinder 0, head 0, record 3 (includes pack catalog disc address). The state of the volume will be unrestricted.

UTCVIM PARAMETERS

Keyword	Parameter	Description
VOL=	n	A 1-6 character volume serial number specifying the IBM disc volume.
PCNAME=	name	A 1-17 character EBCDIC file name specifying the VTOC entry for space saved on the IBM file for the Memorex pack catalog. (Area allocated by IBM programming systems.) Required; default is job termination.
VOLTYP=	{OS DOS}	Specifies the operating system (OS or DOS) that generated the VTOC. Default is DOS.

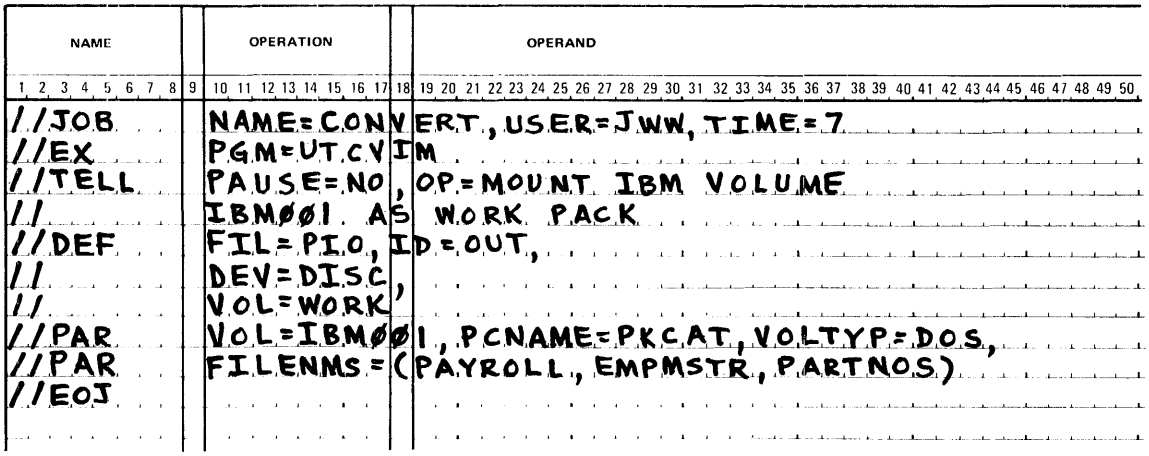
Keyword	Parameter	Description
FILENMS=	(name 1, name 2, . . . , name n)	A 1-17 character file name specifying from 1-15 fixed length record files. The FILENMS parameter is ignored if VOLTYP=OS. Default is NO; no files have the defined characteristics but a pack catalog is to be generated.

UTCVIM REQUIRED CONTROL LANGUAGE

This utility program requires the standard //JOB and //EX statements. A //DEF statement defines the IBM file to be converted. Utility parameters are specified on //PAR cards.

UTCVIM EXAMPLE

The following is an example of the Control Language statements required to convert the IBM DOS system generated disc volume IBM001 to a Memorex disc volume. The VTOC entry for the Pack Catalog space allocated by the IBM programming systems is PKCAT. The files described in the parameter statements have file names PAYROLL, EMPMSTR, and PARTNOS.



```
NAME      OPERATION    OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB     NAME=CONVERT,USER=JWW,TIME=7
//EX      PGM=UTCVIM
//TELL    PAUSE=NO,OP=MOUNT IBM VOLUME
//        IBMØØ1 AS WORK PACK
//DEF     FIL=PIO,ID=OUT,
//        DEV=DISC,
//        VOL=WORK'
//PAR     VOL=IBMØØ1,PCNAME=PKCAT,VOLTYP=DOS,
//PAR     FILENMS=(PAYROLL,EMPMSTR,PARTNOS)'
//EOJ
```

MEMOREX DISC VOLUME TO IBM DISC VOLUME – UTCVMI

This program runs under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTCVMI PROGRAM CAPABILITIES

This utility program converts a Memorex disc volume back to an IBM disc volume; the volume must have originated as an IBM volume and be an output of the Memorex UTCVIM utility program.

CONVERSION PROCEDURE

UTCVMI restores the data to track 0, cylinder 0. This information was saved on an unused track by UTCVIM.

If the output of UTCVMI is to be used by IBM OS, the last record pointer for files that have had records added or deleted by Memorex systems must be updated. These files are specified in the FILENMS parameter.

UTCVMI PARAMETERS

Keyword	Parameter	Description
VOL=	n	A 1-6 character volume identifier of the disc volume to be converted. Required; default is job termination.
VOLTYP=	{ OS DOS }	Specifies the operating system that generated the VTOC. Default is DOS
FILENMS=	(name1,. . ., namen) NO	The name1,. . .,namen entries are 1-17 character names of 1-15 files whose last record pointer must be updated. The FILENMS parameter is ignored if VOLTYP=DOS. Default is NO; no update is necessary.

UTCVMI REQUIRED CONTROL LANGUAGE

The UTCVMI utility program requires the standard //JOB and //EX statements and a //DEF statement for the volume to be converted. Utility parameters are specified in //PAR statements.

UTCVMI EXAMPLE

The following Control Language statements are required to reconvert an IBM disc volume. Only the PAYROLL file has had records added or deleted.

```
NAME              OPERATION           OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB             NAME=CONVERT,USER=JWW,TIME=7
//EX              PGM=UTCVMI
//DEF             FIL=PIO,DEV=DISC,ID=OUT,
//                VOL=WORK
//PAR             VOL=IBMØØ1,VOLTYP=DOS,
//PAR             FILENMS=PAYROLL
//EOJ
```

IBM TAPE FILE TO MEMOREX TAPE FILE – UTCVTM

This program operates under control of MRX/OS. The user supplies Control Language statements to load and execute the program.

UTCVTM PROGRAM CAPABILITIES

This tape conversion program converts an IBM format V tape file into a Memorex tape file. Memorex tape processing is compatible with IBM tape formats, F and U; no conversion of these tapes is necessary.

IBM FORMAT V REQUIREMENTS

- May have standard IBM labels or no labels.
- Input file

  Labeled tape – file is between first 2 tape marks read

  Unlabeled tape – file is from first record to first tape mark (if first record is not a tape mark).
- Input records can be blocked or unblocked, variable or fixed length, with IBM record and block prefixes.

OUTPUT TAPE

- Labeled or unlabeled.
- Blocked or unblocked records – depending upon input tape.
- Memorex common stored data format (IBM record and block prefixes are removed).

UTCVTM PARAMETERS

Keyword	Parameter	Description
RECSZ=	n	The maximum length in bytes of an input record (including the 4 control bytes). Required parameter.

Keyword	Parameter	Description
BLKFT=	n	Specifies the input and output tape blocking factor. Optional; default value is 1. Block size = (RECSZ x BLKFT) + 4.

UTCVTM REQUIRED CONTROL LANGUAGE

The UTCVTM utility conversion program requires the standard //JOB and //EX statements and //DEF statements for both the input IBM tape file and the output Memorex tape file. //PAR statements specify the UTCVTM utility program parameters.

UTCVTM EXAMPLE

The following Control Language statements are required to convert the IBM format V tape file IBMT1 to the Memorex tape file MEMT1. Both input and output records are 80 bytes long and blocked 10 records/block.

```
NAME      OPERATION        OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB    NAME=CONVERT,USER=YOU
//EX     PGM=UTCVTM
//DEF    ID=INP,FIL=I1,
//       DEVICE=TAPE8,VOL=IBMT1,
//       LAB=X,SIZ=8Ø,BLK=1Ø
//DEF    ID=OUT,FIL=M1,
//       DEVICE=TAPE8,VOL=MENT1,
//       LAB=X,SIZ=8Ø,BLK=1Ø
//PAR    RECSZ=8Ø,BLKFT=1Ø
//EOJ
```

EMULATED IBM DISC FILE TO MRX DISC FILE – UTCVEM

This program runs under control of MRX/OS and the Model 20 Compatibility System. The user supplies Control Language Statements and JCL statements to load and execute the program.

UTCVEM PROGRAM CAPABILITIES

The UTCVEM conversion utility program creates a native MRX disc file from an emulated IBM M20 2311 disc file. This utility program requires the minimum MRX/40 or MRX/50 system configuration and an additional tape or disc drive if the user employs an intermediate file during conversion.

UTCVEM can be divided into two modules logically and physically. The first module (UTCVE2) executes in the emulation mode under control of the M20 Compatibility System (emulator). The second module (UTCVEM) executes in the native mode under control of the Memorex Operating System. The two modules are able to pass control to each other via the native-mode exit feature. This feature allows the UTCVEM module to be link edited to the M20 emulator.

UTCVEM is supplied to the user via the Memorex job library. However, UTCVE2 cannot be a part of the Memorex job library, and must be cataloged in the M20 program library to be executed. The program is supplied as a binary deck ready for input to the M20 CMAINT program. It is assumed that any M20 disc file to be converted has previously been transferred to a "Virtual 2311" by use of the IBM utilities BACKUP and RESTORE.

For further information, refer to the **MRX/OS Model 20 Compatibility System Reference** manual.

INPUT/OUTPUT SPECIFICATIONS

Input is a sequential or an indexed sequential simulated M20 disc file. Output can be either a sequential or an indexed sequential disc file.

Input and output files are subject to the following restrictions:

- An input file cannot be defined as Direct* and an output file cannot be defined as relative; both must be direct access.
- Input and output indexed sequential files must have the same key position and key length.
- If the input file is sequential and the output file is indexed sequential, the input file must be in sequence with the key in the position specified in a //PAR statement.
- The optional intermediate output file must be sequential.

*Direct, when capitalized, refers to an IBM definition of an access method; IBM Direct access = MRX random access.

If any of the preceding rules are violated, all open files are closed and the job step is terminated. Error messages are written in either IBM or MRX format. Not all logical errors can be detected at file open time. Therefore, when converting from a sequential file to an index file, it is the user's responsibility to ensure that the input file is sorted and that the key is in the location expected by the output file.

UTCVEM PARAMETERS

Certain file attributes can be specified by the Parameter statement. Attributes may be specified and the conditions under which they may be required are:

Attributes	Condition
input file	Block size must be specified if the file has multi-track blocks.
output file	Record size may be specified if the user wishes to expand the record during conversion.
key position	Key position must be specified if the input file is sequential and the output is indexed. If this condition is not true this parameter is ignored.
key length	Key length may be specified for a sequential to indexed conversion. If the input file is sequential and the user does not specify key length, the length is taken from the output file's label.

All other attributes are taken from the label of the appropriate file. Keywords used to specify parameters in the //PAR card are listed below. The //PAR card is free-format and any number of //PAR cards may be used. However, a keyword and its corresponding parameter must not cross from one card to the next.

Keyword	Parameter	Description
M20BLK=	nnn	The size of the M20 data buffer. This parameter is required for multi-track blocks.
RECSIZ=	nnn	The size of the output record. This parameter is optional. The default is the size of the input record.
KEYLEN=	nnn	The length of the index key. This parameter is required only if converting a sequential file to an indexed sequential file.
KEYPOS=	nnn	The key position in the record. 1 is the first character position. This parameter is required only if converting a sequential file to an indexed sequential file.

UTCVEM REQUIRED CONTROL LANGUAGE

UTCVEM requires //JOB and //EX control statements, a //DEFINE card with ID=OUTPUT for the output file, a //DATA statement (followed by an M20 IPL deck and M20 JCL statements, /* (end of data statement), and an end of job statement.

The bracketed [ ] statements are optional; comments are enclosed by < > symbols.

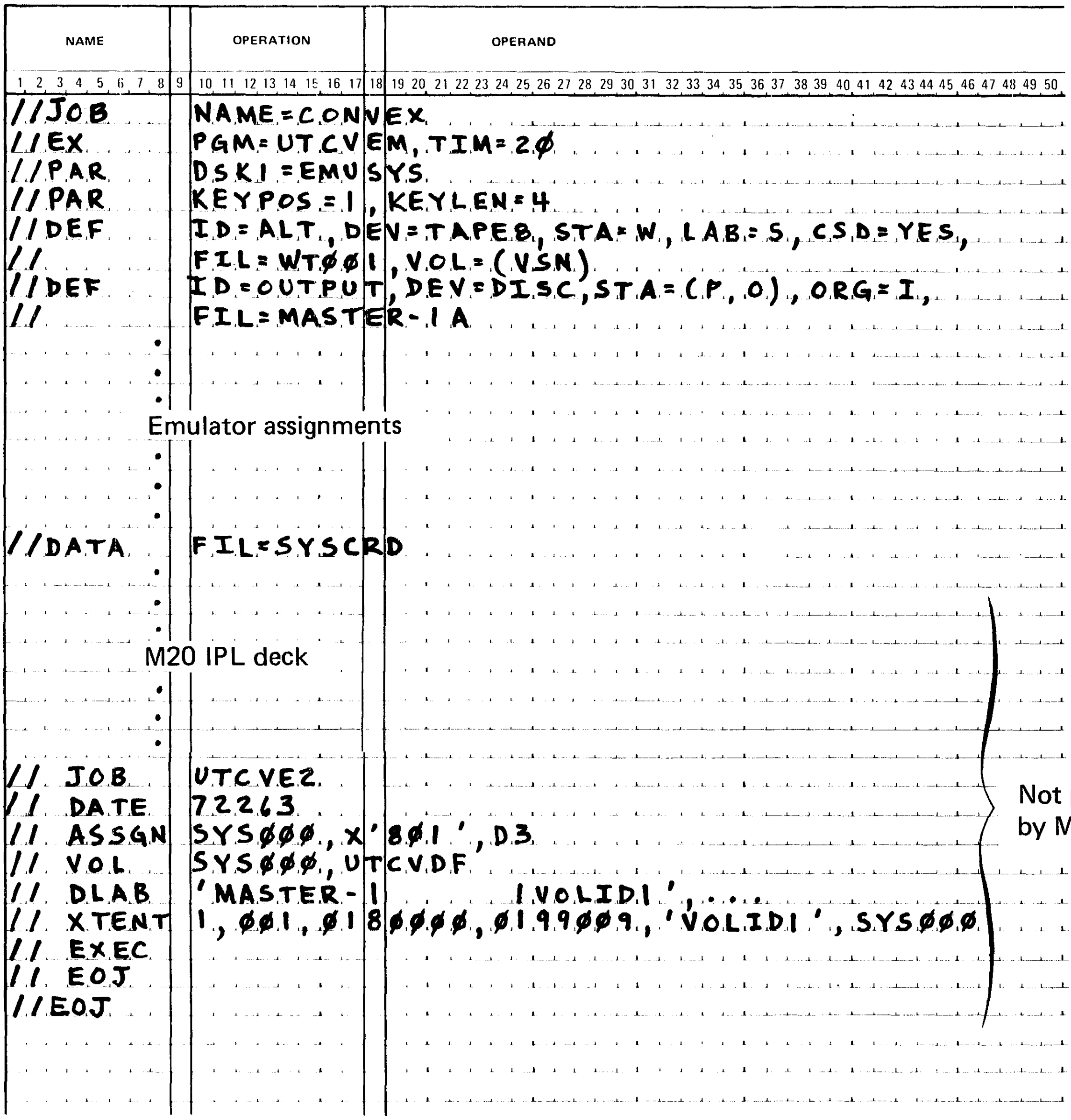
```
//JOB NAM=any-name
//EX PGM=UTCVEM,TIM=60
//PAR DSK1=name of M20 virtual volume (Parameter card for the emulator)
[//PAR M20BLK=nnn,RECSIZ=nnn] }
                              }  (Parameters for UTCVEM)
[//PAR KEYLEN=nnn,KEYPOS=nnn] }
                                      { I }
//DEF ID=OUTPUT,DEV=DISC,STA=(P,O),ORG={   } ,FIL=filename
                                      { S }
                 { DISC  }
[//DEF ID=ALT,DEV={       } ,STA=W,FIL=filename]      (See note below)
                 { TAPE8 }
                          .
                          .
                          .
                          .
(Standard emulator file definitions go here, //DEF)
                          .
                          .
                          .
//DATA FIL=SYSCRD
                          .
                          .
                          .
(An M20 IPL deck goes here)                                   \
                          .                                    |
                          .                                    |
                          .                                    |
//JOB UTCVE2                                                   |
//DATE julian-date                                             |
//ASSGN SYS000,X'801',D3                                       >  Not processed
//VOL SYS000,UTCVDF                                            |  by MRX/OS
//DLAB '<VOLUME LABEL OF M20 FILE>'                            |
//XTENT 1,001,<DATA EXTENT DESCRIPTION>                        |
[//XTENT 2,002,<OVERFLOW EXTENT DESCRIPTION>]                  |
[//XTENT 4,003,<CYLINDER INDEX EXTENT DESCRIPTION>             |
// EXEC                                                        |
// EOJ                                                        /
```

NOTE

If the input and output files are both on the same physical volume, the user may specify file conversion via an intermediate file. This speeds the total conversion time by reducing disc arm movement. The user may specify an intermediate file by providing a //DEF card with ID=ALT. ALT must be a sequential tape or disc file. If ALT is a disc file, it should not be allocated to the same physical volume as the file that is to be converted. The intermediate file must be a native mode Memorex file, not an M20 file.

UTCVEM EXAMPLES

The following example converts a simulated sequential IBM M20 file named MASTER-1 to an index sequential MRX file named MASTER-1A using an intermediate tape file labeled WT001.



NAME | OPERATION | OPERAND

```
//JOB    NAME=CONVEX
//EX     PGM=UTCVEM,TIM=2Ø
//PAR    DSK1=EMUSYS
//PAR    KEYPOS=1,KEYLEN=4
//DEF    ID=ALT,DEV=TAPE8,STA=W,LAB=S,CSD=YES,
//       FIL=WTØØ1,VOL=(VSN)
//DEF    ID=OUTPUT,DEV=DISC,STA=(P,O),ORG=I,
//       FIL=MASTER-1A
        •
        •
        •
     Emulator assignments
        •
        •
        •
//DATA   FIL=SYSCRD
        •
        •
        •
     M20 IPL deck
        •
        •
        •
// JOB   UTCVE2
// DATE  72263
// ASSGN SYSØØØ,X'8Ø1',D3
// VOL   SYSØØØ,UTCVDF
// DLAB  'MASTER-1           1VOLID1',....
// XTENT 1,ØØ1,Ø18ØØØØ,Ø199ØØ9,'VOLID1',SYSØØØ
// EXEC
// EOJ
//EOJ
```

Not processed by MRX/OS

The next example converts a simulated indexed sequential IBM M20 file named PAYROLL directly to a MRX sequential file named PAYROLL. In this example, the input file has a block size of 2400 (80 30-byte characters per block). The output blocking factor is also 80, but the output block size is 2720 because a 4-byte CSD field is included in each output record.

```
NAME       OPERATION         OPERAND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
//JOB     NAME=STREAMI
//EX      PGM=UTCVEM,TIM=35
//PAR     DSKI=EMUSYS
//DEF     ID=OUTPUT,DEV=DISC,STA=(P,O),ORG=S,
//        FIL=PAYROLL,NUM=50ØØ,SIZE=3Ø,
//        BLK=272Ø
       .
       .
       .
   Emulator assignments
       .
       .
       .
//DATA    FIL=SYSCRD
       .
       .
       .
   M20 IPL deck
       .
       .
       .
// JOB    UTCVE2
// DATE   72271
// ASSGN  SYSØØ1,x'8Ø1',D3
// VOL    SYSØØ1,UTCVDF
// DLAB   'PAYROLL           IVOLIDI',....
// XTENT  1,ØØ1,Ø125ØØØ,Ø16ØØØ9,'VOLIDI',SYSØØ1
// XTENT  4,ØØ2,Ø123ØØØ,Ø124ØØ9,'VOLIDI',SYSØØ1
// EXEC
// EOJ
//EOJ
```

Not processed by MRX/OS

DISC FILE CONVERSION UNIT COPY – UTDFCU

The Disc File Conversion Unit Copy utility program (UTDFCU) copies data files from IBM 1316 disc pack files via the Disc File Conversion Unit (DFCU) to MRX 660 disc pack files in a format acceptable to the M20 Compatibility System (Emulator). (See **MRX/OS Model 20 Compatibility System Reference** manual.)

In addition to the requirements listed in Table 1-1, this program requires a DFCU and an IBM 2311 Model 11 (200 cylinders), Model 12 (100 cylinders), or MRX 620 disc drive connected to the DFCU.

The DFCU is a disc controller that enables a MRX system to read from an IBM 1316 disc pack mounted on either an IBM 2311 (Model 11 or 12) or a MRX 620 disc drive. The COPY command requires two disc drives, one for the DFCU and one for SYSRES.

UTDFCU TEST ROUTINES

When UTDFCU is executed, the disc pack copy can be preceded by a series of test routines to ensure proper operation of the DFCU.

UTDFCU test routines are a series of operator selected routines that exercise the DFCU and report malfunctions. The operations are of varying degrees of difficulty and can be executed repeatedly. Each routine performs the specified operation, compares the results with the expected results, and reports discrepancies to the console. The following operation requests are entered through the operator's console. See *UTDFCU Operator Procedures.*

TEST ROUTINE PARAMETERS

In the test routines described in this section, the parameters are entered through the operator's console. When a test command is entered, it will continue until the operation is complete or until another command is entered. Test routine parameters are in the format:

command, seek pattern, repeat factor

Where:

command — Specifies the operation: IFATEST, NORMAL, SKIPTO, RESTORE, SEEK, READHA, READCNT, SRCHHA, SRCHCNT, READSEC, READTRK, VFYSEC, VFYTRK, DUMP, EOJ.

seek pattern — Specifies a variable seek pattern or a specific disc address:

RSP – (random seek pattern) the routine is executed 203 times on random addresses.

BSP – (backwards seek pattern) the routine is executed on each cylinder and head starting with cylinder 202 head 9 and working backwards through cylinder 0 head 0.

FSP — (forward seek pattern) the selected routine is executed on each cylinder and head starting with cylinder 0 head 0 and proceeding through cylinder 202 head 9.

ASP — (all seek patterns) the routine is executed through RSP, BSP, and FSP.

If the seek pattern is a specific address, it is in the format:

    cch[r]

where:

cc   is cylinder number ($00\text{-}CA_{16}$)

h   is head number ($0\text{-}A_{16}$)

r   is record number

A specific address must be in hexadecimal.

repeat factor   Specifies whether the command sequence is to be executed only one time or repeated continuously.
C indicates that the command is to be repeated continuously.
The default is 1; the command will be executed once.

IFATEST

The IFATEST format is:

    IFATEST

The IFATEST mode of operation provides for testing the generation of home addresses, count fields, cyclic checks, and inter-record gaps. UTDFCU generates home address and count fields corresponding to the last disc address seeked. All of the UTDFCU operations except VFYSEC and VFYTRK can be performed in IFATEST mode. To return to the normal operation mode, enter the NORMAL command.

NORMAL

The NORMAL format is:

    NORMAL

This command resets UTDFCU to a normal mode of operation. If the program is not operating in IFATEST mode, the command is ignored.

SKIPTO

The SKIPTO format is:

```
SKIPTO
```

This command specifies that UTDFCU is to bypass all error typeouts. SKIPTO applies only to the test in progress when it is issued.

RESTORE

The RESTORE format is:

```
RESTORE [,,repeat factor]
```

This routine performs a disc restore operation and checks for proper ending status.

SEEK

The SEEK format is:

```
SEEK, seek pattern [,repeat factor]
```

This routine performs a seek to the specified disc address(es) and checks for proper ending status after each seek.

READHA

The read home address format is:

```
READHA, seek pattern [,repeat factor]
```

The READHA routine seeks the specified disc address, reads the home address, and checks for the proper ending status. If the proper ending status is received, the home address read in is compared to the disc home address and any differences are reported.

READCNT

The read count field format is:

```
READCNT
```

The READCNT routine seeks the specified disc address, reads 10 count fields (sectors), and checks for the proper ending status. If the proper ending status is received, the count field address portion is compared to the disc address seeked and the count field data length portion is checked for a byte length of 270 – the 2311 disc sector byte length.

SRCHHA

The search home address format is:

SRCHHA, seek pattern [,repeat factor]

This routine seeks the specified disc address and performs a search home address for the disc address seeked.

SRCHCNT

The search count field format is:

SRCHCNT, seek pattern [,repeat factor]

This routine seeks the specified disc address and checks for the specified record number. If no record number is specified, the routine checks for each of the 10 count fields on each track.

READSEC

The read sector format is:

READSEC, seek pattern [,repeat factor]

This routine seeks the specified disc address, performs a search count operation, and reads 270 data bytes. If no record number is specified in the seek pattern, the search count operation is performed on each of the 10 sectors on each track.

READTRK

The read track format is:

READTRK, seek pattern [,repeat factor]

This routine seeks the specified disc address, performs a search home address operation, and reads the 10 sectors. Each sector is compared with the disc address seeked, and the operator is notified if they do not match.

VFYSEC

The verify sector data format is:

VFYSEC, seek pattern [,repeat factor]

This routine requires a test pack that has the following 270-byte data pattern:

1. Each sector's disc address (cchhr) repeated 53 times

2. 5 bytes of X'FF'

VFYSEC seeks the specified disc address, performs a search count operation followed by a read data command. The routine compares the data sector disc address (cchhr) with the data in the buffer and notifies the operator of differences. If the record number is not specified, the data sector verification routine is performed on each of the 10 sectors on the track.

VFYTRK

The verify track data routine format is:

```
VFYTRK, seek pattern [,repeat factor]
```

This routine requires a test pack that has the following 270-byte data pattern:

1. Each sector's disc address (cchhr) repeated 53 times
2. 5 bytes of X'FF'

VFYTRK seeks the specified disc address and executes a channel program; a search home address, and 10 read count and read data commands. The routine then compares the data in each sector buffer with its corresponding sector address (cchhr) and notifies the operator if there are differences.

DUMP

The format of the dump command is:

```
DUMP,cch
```

Execution of this command results in program termination. The command dumps the DFCU buffer – 3072 bytes including inter-record gap data as well as the 10 count and data fields of a track. DUMP seeks the specified disc address and reads home address with a data length of 3072 bytes rather than 5 bytes. This generates the following console display:

- Burst check error
- Starting address of the DFCU
- DFCU dump

EOJ

The format of the end of job command is:

```
EOJ
```

This command terminates the UTDFCU program.

COPY

The format of the copy utility command is:

COPY, volsn, filename

Where:

volsn — Specifies the 6 character volume serial number of the 1316 pack to be copied. The COPY routine verifies this entry. An optional parameter.

filename — Specifies the 1-17 character name of the receiving file; filename must be the same as that specified in the //DEF statement with ID=EMUFILE.

This command effects a copy of a complete 1316 pack to a MRX 660 disc pack. The output is written in the Emulator Virtual format. Four 1316 cylinders are written on one MRX 660 cylinder. The input packs can be written on either a MRX 2311 Model 11 drive or a MRX 2311 Model 12 drive. The size of the receiving file determines how many cylinders will be copied. The number of tracks copied from the input file equals the number of blocks copied to the output file.

RECEIVING FILE ALLOCATION – NUMBER OF BLOCKS*

The receiving file may be allocated either prior to running UTDFCU or by CLS at run time. It must start on a cylinder boundary and be allocated on a contiguous block of mass storage.

	Input Disc Attributes	
Input Disc Pack Model	Cylinders 1-3 Reserved for Alternate Tracks	No Alternate Track Region Reserved; Input Disc Can Have No Defective Tracks
IBM Model 12	500 blocks Cylinders 0 and 4-102 are copied.	520 blocks Cylinders 0-102 are copied.
IBM Model 11	1000 blocks Cylinders 0 and 4-202 are copied.	1020 blocks Cylinders 0-202 are copied.

*See the NUM= parameter of the //DEF statement with ID=EMUFILE.

UTDFCU OPERATING PROCEDURE

START UP PROCEDURE

1. Set the ENABLE/DISABLE switch to DISABLE.
2. Position the unit select rotary dial to the required unit number before replying Y to the MRX/OS mount WORK message.

UTDFCU OPERATOR COMMUNICATION

When the program is loaded, it displays the message:

```
DFCU READY – ENTER COMMAND
```

The operator enters a test or copy command. If the command is valid, it is placed into operation. Otherwise the message:

```
INVALID COMMAND – TRY AGAIN
```

is displayed and the operator must enter a command. When the test commands are placed into operation, the following message is displayed:

```
DFCU READY FOR NEXT COMMAND
```

The operator can now change the test selection. When the COPY command is entered, a message is sent containing the input volume serial number, number of tracks to be copied, receiving filename, and volume identifier.

If the copy is unsuccessful, the message, COPY ABORTED is displayed. When the copy is complete, UTDFCU sends the message:

```
COPY COMPLETE
```

and the job terminates normally.

UTDFCU REQUIRED CONTROL LANGUAGE

UTDFCU requires the standard //JOB, //EXEC, and //EOJ statements. A //DEF statement with ID=DFCU, FILE=PIO, VOL=WORK is always required for the DFCU. A //DEF statement with ID=EMUFILE is required for the receiving file if a copy is performed; the requirements for this //DEFINE statement depend on whether the receiving file has been allocated.

CONTROL LANGUAGE EXAMPLES

To allocate the receiving file during UTDFCU execution, use the following CLS:

```
//JOB    NAME=RUN1
//EX     PGM=UTDFCU
//DEF    ID=DFCU,FIL=PIO,VOL=WORK
                                                ⎧ 500,⎫
//DEF    ID=EMUFILE,FIL=filename,VOL=volid,NUM= ⎨ 520, ⎬,
                                                ⎪1000,⎪
                                                ⎩1020,⎭
//       SIZ=5400,STA=P,LOC= ⎧YES⎫ ,CON=YES,ORG=R,
                             ⎩nnn⎭
//       CSD=NO[,CAT=NO]
//EOJ
```

To execute UTDFCU with a previously allocated receiving file, use the following CLS:

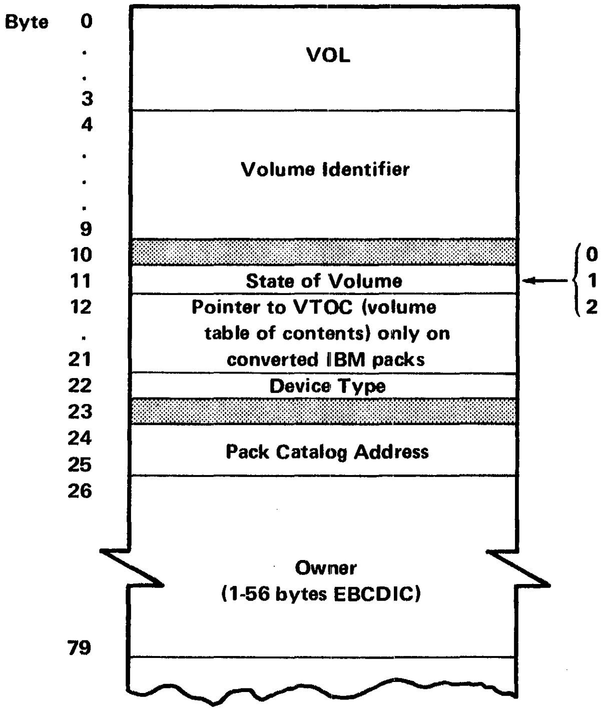
```
//JOB    NAME=RUN2
//EX     PGM=UTDFCU
//DEF    ID=DFCU,FIL=PIO,VOL=WORK
//DEF    ID=EMUFILE,FIL=filename,VOL=volid,STA=P
//EOJ
```

A. DISC AND TAPE LABELS

DISC LABELS

Disc packs have a standard volume label containing volume identification and owner information. The volume label for each disc pack is located at cylinder 0, head 0, record 3.

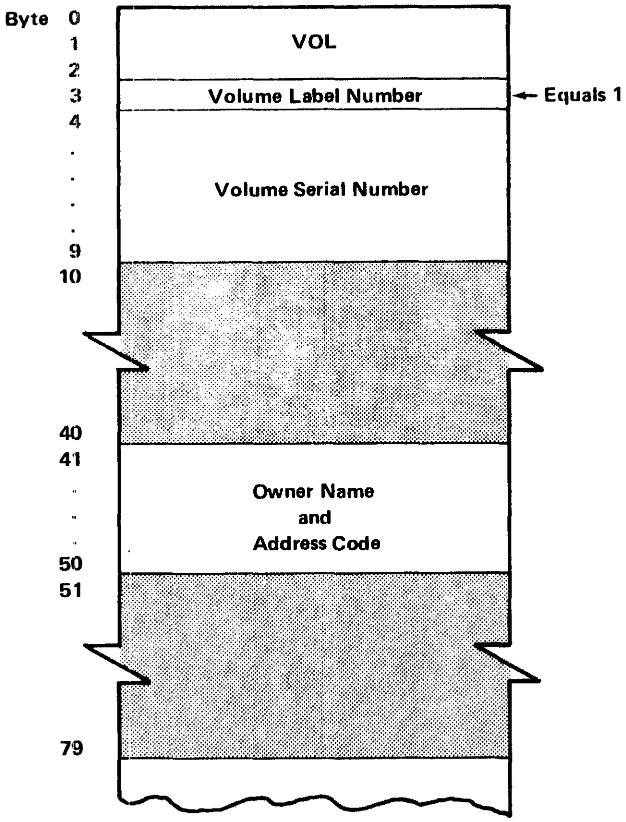
The device type, owner, and state of the volume – unrestricted (0), restricted (1), or locked (2) – are included along with the starting track address of the pack catalog for this volume. The actual file identification is found in the disc catalogs.



TAPE LABELS

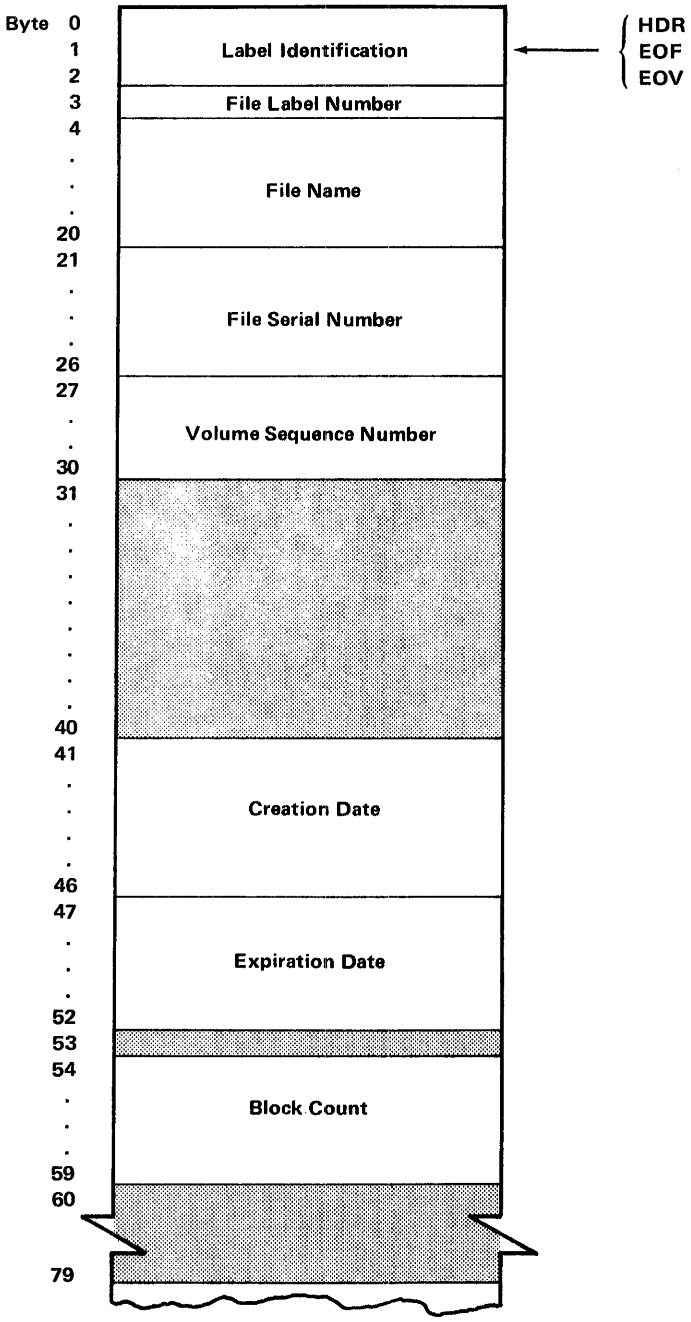
STANDARD TAPE VOLUME LABEL

The volume label is located at the beginning of a tape reel and is identified by the characters, VOL, found in the first three positions. The volume label number is always 1 for compatibility with IBM. The volume serial number occupies positions 4 through 9 and identifies the volume. A unique owner name and address code identifies the installation.



STANDARD TAPE FILE LABEL

The standard tape file label provides information concerning the user's file such as creation and expiration dates, file name, and sequence number. The label identification field identifies the type of standard label with a three letter abbreviation. Three types of labels supported by the system are header labels (HDR), end-of-file labels (EOF), and end-of-volume labels (EOV). The file label number found in byte 3 is 1 or 2. The file serial number, found in positions 21-26, is identical to the volume serial number in the volume label of the first volume. The volume sequence number identifies the order of the volume of data records in a multivolume logical file. The block count provides the number of physical records written in a file at creation.



B. SUMMARY OF CONTROL LANGUAGE STATEMENTS

JOB LEVEL STATEMENTS

Within a job, all Control Language statements are allowed. The following are required:	
//JOB	First statement of every job.

//EXECUTE
//EXEC
//EX
	The first command following the //JOB statement, whether in line or as the first executable command (following //DECLARE) of a called procedure.

//EOJ	Last statement of every job.

STEP LEVEL STATEMENTS

//EXECUTE
//EXEC
//EX
	The first executable statement of every step, identifies the program to be executed.

The following statements may also be included:	
//PAR	Specifies run-time parameters to the program. If a statement requires multiple cards, every parameter except the last parameter on the last //PAR card is followed by a comma.
//DEFINE //DEF	Specifies devices, volumes, and files requested by the step. See individual utility programs for ID parameter requirements.
//ROUTE //RTE	Allows output spooling or unit record device allocation.
//SET	Specifies job date and/or POST byte switch settings.
//TELL //TEL	Places messages on the operator's console.
//CALL	Merges Control Language statements from a cataloged procedure.
*	Comment statement.
Within a step, the following statements are not allowed:	
//JOB	First statement of every job.
//DATA //DAT	Defines the following data file. If data is spooled, it may appear anywhere between //JOB and //EOJ. If not spooled, it must immediately precede the //EOJ.
/*	Specifies the end of a card file.
//DECLARE //DEC	Designates the following control statements as a cataloged procedure, provides the specification statement for the procedure.
//IF	Provide for branching based on condition code test.
//EOJ	Last statement of every job.

INTERSTEP LEVEL STATEMENTS

The only Control Language Statements occurring between steps are:	
//IF	Provide for forward branching based on condition code test.
*	Comment statement.

CATALOGED PROCEDURES

Within a cataloged procedure, the following control language statement is required:	
//DECLARE //DEC	Specifies all keywords to be provided in calling the procedure; required as first statement of a cataloged procedure.
The following statements may also be included:	
//EXECUTE //EXEC	The first executable statement of every step, identifies the program to be executed.
//PAR	Specifies run-time parameters to the program.
//DEFINE //DEF	Specifies devices, volumes, and files requested by the step.
//SET	Specifies job date and/or POST byte switch settings.
//TELL //TEL	Places messages on the operator's console.
//IF	Provide for forward branching based on condition code test.
*	Comment statement.
The following statements are not allowed:	
//JOB	First statement of every job.
//DATA //DAT	Identifies the following data file.
/*	Specifies the end of a card file.
//CALL //CAL	Calls a cataloged procedure.
//EOJ	Last statement of every job.

DATA LEVEL STATEMENTS

A data file identified with the CLS=YES keyword on its //DATA statement may contain any Control Language statement, except /* which terminates all data files. If CLS=NO on the //DATA statement, any Control Language statement except *(comment), will terminate the data file. CLS=YES has no meaning with FIL=SYSCRD.

C. DISC INITIALIZE CYLINDER 0 UPDATE DECK

The cylinder 0 update deck may have 1-7 machine loadable object decks:

- Autoload deck – preceded by a card with AL in columns 1 and 2
- Control storage deck – preceded by a card with CS in columns 1 and 2
- 1-5 diagnostic decks – each deck preceded by a card with D1, D2, D3, D4, or D5 in columns 1 and 2 (D1 card precedes the first deck, D2 card precedes the second deck. . .)

Any one of these groups of decks (autoload, control storage, or diagnostic) can be loaded in any order. However, the diagnostic decks must be loaded as a group. Each deck is followed by a data delimiter card (/* in columns 1 and 2). Two data delimiters terminate the cylinder zero update deck (see example in Figure C-1).

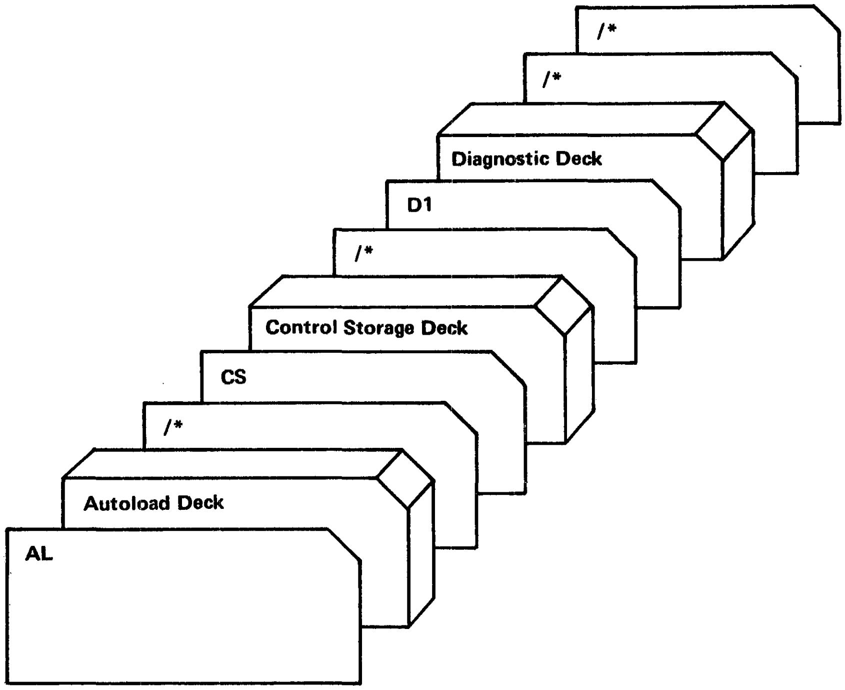


Figure C-1. Cylinder 0 Update Deck Example

D. OPERATOR PROCEDURES FOR STAND-ALONE UTILITY PROGRAMS

This appendix lists the operator procedures required to run Memorex utility programs that do not run under control of the Memorex Operating System.

These utility programs are:

- Disc Initialize
- Stand-Alone Disc-to-Disc Copy
- Stand-Alone Memory Dump

Refer to the Disc Initialize and Stand-Alone Disc-to-Disc Copy program descriptions in Section 2 for lists of parameters requested by these programs.

When responding to console requests, do not press the INT key before responding. The stand-alone utilities differ from the system dependent utilities in this respect.

LOADING MICROCODE FROM DISC

The Disc Initialize and Disc-to-Disc Copy stand-alone utilities require microcode in control storage. The Memory Dump utility program runs in control storage; microcode must be reloaded after executing the Memory Dump.

To load microcode into control storage:

1. Mount the system disc pack on a disc drive.
2. Switch the disc drive power switch on and set the disc mode switches to ENABLE and READ/WRITE.
3. Ensure that the mode toggle switches on the operator's console are set to COMM and HALF DUPLEX.
4. Switch on and make ready all input/output devices.
5. Ensure the following conditions on the System Control Panel:
    a. The MAINTENANCE MODE light is off.
    b. The PROGRAM MODE light is off.
    c. The LOCAL/REMOTE switch is in the LOCAL position.
    d. The LOAD SELECT switch is in the PRIMARY of DISC position.

6. Press the POWER ON pushbutton.

7. When the POWER OFF light is extinguished, press the RESET/LOAD pushbutton.

When control storage and the systems are loaded, the following messages are typed on the operator's console:

```
SYSTEM LOADED – VERSION n.n

P1 I ENTER TIME AND DATE

1 P1 D RETAIN JOB QUE?
```

See the **MRX/OS Operating Procedures** manual for more information on these messages and operator response.

DISC INITIALIZE OPERATOR PROCEDURE

Before executing the Disc Initialize utility program, the disc pack to be initialized must be mounted for READ/WRITE.

CLEARING MAIN STORAGE

To write zeros in main storage:

1. Ensure that either the PROGRAM MODE or the MAINTENANCE MODE pushbutton is lit.

2. Set the CONSOLE ADDRESS REGISTER SELECT rotary switch to ADDRESS.

3. Set the CONSOLE DATA REGISTER SELECT switch to DATA.

4. Position the CONSOLE MODE SELECT rotary switch to MS-WR.

5. Press SYSTEM RESET.

6. Position CONSOLE CONTROL SELECT switch to NORMAL.

7. Press CONSOLE RUN.

8. Position CONSOLE CONTROL SELECT switch to STOP/STEP.

EXECUTION

To execute the Disc Initialize utility program:

1. Ensure that microcode is loaded. (Events such as a control storage parity error or the execution of the Stand-Alone Memory Dump utility program may require a microcode reload at this point. See **MRX/OS Operating Procedures** manual.)

2. Press the MAINTENANCE MODE pushbutton on the System Control Panel.

3. Load the UTSSDI object deck into main storage at address zero.

   a. Press SYSTEM RESET. (This insures a load at main storage address zero.)

   b. Position CONSOLE MODE SELECT selector to MS-WR.

   c. Position LOAD SELECT switch to the up position (ALTERNATE) (either ALTERNATE or CR).

   d. Place the UTSSDI object deck in card reader and press card reader START.

   e. Press RESET/LOAD.

*4. Set processor 4 P-register to zero (after card deck is loaded).

   a. Press SYSTEM RESET.

   b. Position CONSOLE MODE SELECT switch to RF-WR.

   c. Position CONSOLE CONTROL SELECT switch to STOP/STEP.

   d. Position CONSOLE ADDRESS REGISTER SELECT switch to ADDRESS.

   e. Position CONSOLE DATA REGISTER SELECT switch to DATA.

   f. Press the CONSOLE ADDRESS REGISTER DISPLAY pushbutton/indicators (08, 12, 15) to enter the hexadecimal value 00089 in the CONSOLE ADDRESS REGISTER.

*UTSSDI can be restarted at this point. It is not necessary to reload.

g. Press the CONSOLE RUN pushbutton.

h. Press SYSTEM RESET.

5. Set PROCESSOR SELECT rotary switch to 4.

6. Press PROCESSOR RUN.

The console responds with a header message and begins the series of parameter request messages.

Example:

```
DISC INITIALIZE  REV. mm-dd-yy
PARAMETERS
DRIVE=
```

After each request for a parameter, the operator enters the required parameter or a carriage return. A carriage return causes the default for a particular parameter. If less than the maximum number of parameter characters is entered, use carriage return after the last character to proceed to the next character. Otherwise, do not use carriage return. UTSSDI then validates the entry. If the parameter is valid, the program proceeds to the next parameter request. If the parameter is invalid, UTSSDI displays the message INVALID ENTRY and repeats the parameter request.

Console parameter requests continue until every UTSSDI parameter has been supplied or until the operator enters ETX. The default values are then assumed for the remaining parameters.

When the utility has completed execution, an end-of-program message is printed on the console stating volume number, drive number, and alternate track assignments.

The utility begins requesting parameters again; another volume can be initialized. If the Disc Initialize has not been restarted, parameter default values are now equal to the parameter entries for the volume just initialized.

ERROR MESSAGES

See Appendix E for a complete UTSSDI error message listing. Some of the console messages request information from the operator. For example:

Message	Operator Response*
DEVICE LABEL EXISTS WITH VOLUME IDENT = nnnnnn CONTINUE?	If nnnnnn = DDDDDD an abnormal termination on a previous initialization occurred. volid DDDDDD cannot be referenced. Reply Y or YES to re-initialize the pack.

*(Do not press the INT key before responding.)

Message	Operator Response*
CARD READ FAILURE RETRY?	Required cards are not in the reader. If CYLO=NO, either the cylinder 0 update deck or a data delimiter card (/*) must be in the card reader. Put the required cards in the reader, press card reader START, and reply Y or YES.

STAND-ALONE DISC-TO-DISC COPY OPERATOR PROCEDURE

Before executing the Disc-to-Disc Copy program, the disc pack to be copied must be mounted for READ ONLY. The pack to be copied to must be mounted for READ/WRITE.

CLEARING MAIN STORAGE

It is recommended that main storage be cleared before loading this utility program to avoid the possibility of program malfunction due to extraneous data remaining in main storage.

To write zeros in main storage:

1. Ensure that either the PROGRAM MODE or the MAINTENANCE MODE pushbutton is lit.
2. Set the CONSOLE ADDRESS REGISTER SELECT rotary switch to ADDRESS.
3. Set the CONSOLE DATA REGISTER SELECT switch to DATA.
4. Position the CONSOLE MODE SELECT rotary switch to MS-WR.
5. Press SYSTEM RESET.
6. Position CONSOLE CONTROL SELECT switch to NORMAL.
7. Press CONSOLE RUN.
8. Position CONSOLE CONTROL SELECT switch to STOP/STEP.

EXECUTION

To execute the Disc-to-Disc Copy utility program:

*(Do not press the INT key before responding.)

1. Ensure that microcode is loaded. (Events such as a control storage parity error or the execution of the Stand-Alone Memory Dump utility program may require a microcode reload at this point. See the **MRX/OS Operating Procedures** manual.)

2. Press the MAINTENANCE MODE pushbutton on the System Control Panel.

3. Load the UTSSDD object deck into main storage at address zero.

   a. Press SYSTEM RESET. (This insures a load at main storage address zero.)

   b. Position CONSOLE MODE SELECT selector at MS-WR.

   c. Position LOAD SELECT switch to the up position (either ALTERNATE or CR).

   d. Place the UTSSDD object deck in the card reader and press card reader START.

   e. Press RESET/LOAD.

*4. Set processor 4 P-register to zero.

   a. Position CONSOLE MODE SELECT switch to RF-WR.

   b. Position CONSOLE CONTROL SELECT switch to STOP/STEP.

   c. Position CONSOLE ADDRESS REGISTER SELECT switch to ADDRESS.

   d. Position CONSOLE DATA REGISTER SELECT switch to DATA.

   e. Press SYSTEM RESET.

   f. Press the CONSOLE ADDRESS REGISTER DISPLAY pushbutton/indicators to enter 00089 in the CONSOLE ADDRESS REGISTER.

   g. Press the CONSOLE RUN pushbutton.

   h. Press SYSTEM RESET.

5. Set the PROCESSOR SELECT rotary switch to 4.

6. Press PROCESSOR RUN.

*UTSSDD can be restarted at this point. It is not necessary to reload.

The console responds with a header message and begins the series of parameter request messages.

Example:

```
DISC-TO-DISC Version n
FROM DRIVE=
```

See **UTSSDD Parameters** in Section 2 for operator response to parameter requests. After the operator enters a volume identifier and its drive number, the utility program reads the device label and makes a comparison check. A mismatch causes rejection of the parameter, console printout of the device label volume number, and a return to reissue the drive number request.

If any parameter is in error, the program sends the message INVALID REPLY and then reissues the request for information.

To halt UTSSDD execution (copy terminates at the end of a track) after the parameters have been accepted:

1. Press the INT (or BREAK) key. YOUR MESSAGE? is written on the console.

2. Reply H or HALT (followed by a carriage return or ETX).

Copy terminates at the end of a track. The program writes the message:

```
DISC-TO-DISC COPY HALTED AS YOU REQUESTED.
```

Return to Step 4 of the procedure for executing the Disc-to-Disc Copy. See Appendix E for a complete UTSSDD error message listing. Some of the console messages request information from the operator. For example:

Message	Operator Response*
YOUR MESSAGE?	To halt execution, reply H or HALT (followed by a carriage return or ETX). To continue normal processing, reply with a carriage return or ETX.
CONTINUE?	To ignore the error condition, reply Y or YES. To halt the program, reply N or NO.
RETRY?	The utility program has already retried 4 times. If the operator replies Y or YES, the program retries 4 more times. If the operator replies NO, the program is terminated. At normal copy termination,

*Do not press the INT key before responding.

Message	Operator Response*
	UTSSDD sends the message: MORE? If there is another pack to be copied, reply Y or YES. The default is NO, which yields the message DISC-TO-DISC COMPLETE.

MEMORY DUMP OPERATOR PROCEDURE

Use one of the two following operator procedures to execute the Memory Dump utility program. Procedure 1 is shorter but clears all 8 P-micro registers, the control register, the busy/active register and the tie breaker register. Procedure 2 is more complex but preserves the register contents.

PROCEDURE 1

1. Press the PROGRAM MODE pushbutton. If the pushbutton will not light, check that the MAINTENANCE MODE pushbutton is not lit and try again.
2. Press SYSTEM RESET pushbutton.
3. Position the LOAD SELECT switch to the up position (either ALTERNATE or CR).
4. Place the UTSSMD object deck in card reader and press card reader START.
5. Press RESET/LOAD pushbutton.
6. After card deck is loaded, set the PROCESSOR SELECT rotary switch to the processor to which the line printer is attached (usually processor 2).
7. Press SYSTEM RESET pushbutton.
8. Ready printer – must be on-line.
9. Press PROCESSOR RUN.

After running the Stand-Alone Memory Dump utility program, the user must reload microcode. (See **Loading Microcode from Disc** at the beginning of this appendix.)

*(Do not press the INT key before responding.)

PROCEDURE 2

1. Press the PROGRAM MODE pushbutton. If the pushbutton will not light, check that the MAINTENANCE MODE pushbutton is not lit and try again.

2. Load the UTSSMD object deck into control storage starting at address zero.

   a. Place all PROCESSOR CONTROL SELECT toggle switches in the STOP/STEP (up) position.

   b. Set BREAKPOINT ADDRESS SELECT rotary switches to address zero.

   c. Set the CONSOLE MODE SELECT rotary switch to the CS-RD position.

   d. Place the CONSOLE CONTROL SELECT toggle switch in the BREAKPOINT (down) position.

   e. Set the CONSOLE ADDRESS REGISTER SELECT rotary switch to S-micro.

   f. Set the CONSOLE DATA REGISTER SELECT rotary switch to DATA.

   g. Press the CONSOLE RUN pushbutton. The S-micro Register, indicated in the CONSOLE ADDRESS REGISTER DISPLAY, should now contain all zeros.

   h. Position CONSOLE CONTROL SELECT switch to STOP/STEP.

   i. Position CONSOLE MODE SELECT switch to the CS-WR position.

   j. Position the LOAD SELECT switch to the position occupied by the Autoload reader (either ALTERNATE or system input device).

   k. Position all PROCESSOR CONTROL SELECT toggle switches in the NORMAL position.

   l. Place the UTSSMD object deck in card read and press card reader START.

3. After card deck is loaded, set the PROCESSOR SELECT rotary switch to the processor to which the line printer is attached (usually processor 2).

4. Clear the P-micro register for the processor selected in step 3.

   a. Set the CONSOLE MODE SELECT rotary switch to RF-WR.

   b. Set the CONSOLE CONTROL SELECT toggle switch to STOP/STEP.

   c. Set the CONSOLE ADDRESS REGISTER SELECT rotary switch to ADDRESS.

   d. Set the CONSOLE DATA REGISTER SELECT switch to DATA.

   e. Press the CLEAR ADDRESS and CLEAR DATA pushbuttons.

   f. Press the CONSOLE ADDRESS REGISTER DISPLAY pushbuttons to enter one of the following values:

Hexadecimal Address to be Entered	Processor Selected in Step 3
00121	1
00141	2

   g. Press CONSOLE RUN pushbutton.

   h. Set CONSOLE MODE SELECT rotary switch to OFF.

5. Ready printer -- must be online.

6. Press PROCESSOR RUN pushbutton.

After running the Stand-Alone Memory Dump utility program, the user must reload microcode. See *Loading Microcode From Disc* at the beginning of this appendix.

UTSSMD PROGRAM STOPS

If any of the UTSSMD program stops occurs, the dump may be retried. In some cases, register contents may have been altered. The stop number appears in extended register 1 (ignoring register bits 0 and 1) of the processor state that the dump routine is being run in.

Stop Number	Meaning of Stop
11	Bounds error (should not happen).
1D	Main storage parity error.*
29	Control storage parity error.
69	Executing processor not 1 or 2.
114	Autoload attempted by user.
3A7, 3A8	Dump complete.

The following stops should not occur. The user may retry the dump after ascertaining that the executing processor is not in stop mode, the correct processor is selected, and the printer is online and ready.

Stop Number	Meaning of Stop
427	No response to select out on channel (including channel not there on processor, 1 or 2).
42D	Channel control check.
434	Channel transmission check.
438	Status in fails to drop as a response to select out drop in control unit busy sequence.
444	Select in fails to drop as a response to select out drop in device offline sequence.
451	Neither printer 0E nor printer 1E are online and connected.
463	Address in fails to raise as a response to drop of address out.
478	Address of responding device not address of selected device.
484	Address in fails to drop as a response to raise of command out.
492	Status in fails to raise as a response to drop of command out.
4AD	Status in fails to drop as a response to raise of service out.
4C9	Request in fails to raise for presentation of device end status.
4DA	Service in or status in fails to raise as a response to drop of service out.**

*A main storage parity on an exact 8K boundary of memory is interpreted as the end of a less than 65K system dump, and the dump completion stop will occur.

**The printer is not ready. If the user has two printers: either the control unit for 0E is on and printer 0E is not ready or the control unit for 0E is off but printer 1E is not ready.

Stop Number	Meaning of Stop
4E5	Service in fails to drop as a response to raise of service out.
4F1	Service in fails to drop as a response to raise of command out.

DISC-TO-TAPE DUMP OPERATOR PROCEDURE

Before executing this utility program, the disc pack to be dumped must be mounted for READ ONLY. The tape write ring must be inserted in the output tape reel before it is loaded.

EXECUTION

To execute the Disc-to-Tape Dump utility program:

1. Ensure that microcode is loaded. (Events such as a control storage parity error or the execution of the Stand-Alone Memory Dump utility program may require a microcode reload at this point. See the **MRX/OS Operating Procedures** manual.)

2. Press the MAINTENANCE MODE pushbutton on the System Control Panel.

3. Load the UTSSDT object deck into main storage at address zero.

   a. Press SYSTEM RESET. (This insures a load at main storage address zero.)

   b. Position CONSOLE MODE SELECT selector at MS-WR.

   c. Position LOAD SELECT switch to the up position (either ALTERNATE or CR).

   d. Place the UTSSDT object deck in the card reader and press card reader START.

   e. Press RESET/LOAD.

*4. Set processor 4 P-register to zero.

   a. Position CONSOLE MODE SELECT switch to RF-WR.

   b. Position CONSOLE CONTROL SELECT switch to STOP/STEP.

   c. Position CONSOLE ADDRESS REGISTER SELECT switch to ADDRESS.

   d. Position CONSOLE DATA REGISTER SELECT switch to DATA.

   e. Press SYSTEM RESET.

   f. Press the CONSOLE ADDRESS REGISTER DISPLAY pushbutton/indicators to enter 00089 in the CONSOLE ADDRESS REGISTER.

   g. Press the CONSOLE RUN pushbutton.

   h. Press SYSTEM RESET.

5. Set the PROCESSOR SELECT rotary switch to 4.

6. Press PROCESSOR RUN.

The console displays a header message and begins the series of parameter request messages. See *UTSSDT Parameters* for operator response to parameter requests. See Appendix E for a complete UTSSDT error message listing.

UTSSDT PROGRAM STOPS

The following program stop numbers can appear in the processor 4 P-register. When the error has been corrected, the program must be restarted at address zero (step 4 of operating procedures).

Stop Number	Meaning of Stop
0002	A branch to this address has been preceded by the console message: TAPE ERROR. . . RETRY? The branch takes place immediately if the user replies N to the message.

*UTSSDT can be restarted at this point. It is not necessary to reload.

Stop Number — Meaning of Stop

If, however, the user replies Y, a branch to this address occurs only if an attempt to backspace the tape in order to retry a write has failed.

The error may be caused by a defective tape or a defective tape drive. Correct the error and restart the Utility.

0004 — A branch to this address means that a modem or disconnect error has occurred for the operator's console. Correct the error and restart the utility.

0006 — A branch to this address occurs when the user replies N to the message:

$\left\{\begin{matrix}\text{DISC}\\ \text{TAPE}\end{matrix}\right\}$ UNSOLICITED ATTENTION
CONTINUE?

0008 — A branch to this address is preceded by the message:

DEVICE cuu TIMED OUT

During the dump this indicates that there is a hardware malfunction; specifically, channel-end is not being returned within a specified amount of time. When this condition occurs, the utility does not send any other message before branching to this address.

During parameter solicitation, however, the utility branch to this address occurs after the time-out error message only if the user has replied N to a subsequent RETRY? . The same class of hardware malfunction is implied.

Recommended action: Take a memory dump and forward it with the console log to your Field Service Representative.

000A
thru
0012 — Stops at these addresses, should not occur.

Recommended action: Dump memory and forward the dump and the console log to the Field Service Representative.

0014 — A branch to this address is caused by an irrecoverable disc search failure. This indicates that either the disc is defective or the drive is malfunctioning.

TAPE-TO-DISC RESTORE OPERATOR PROCEDURE

Before executing this utility program, the first tape from which the disc will be restored must be loaded without a write ring. The disc pack must be mounted for READ/WRITE.

EXECUTION

To execute the Tape-to-Disc Restore utility program:

1. Ensure that microcode is loaded. (Events such as a control storage parity error or the execution of the Stand-Alone Memory Dump utility program may require a microcode reload at this point. See the **MRX/OS Operating Procedures** manual.)

2. Press the MAINTENANCE MODE pushbutton on the System Control Panel.

3. Load the UTSSTD object deck into main storage at address zero.

   a. Press SYSTEM RESET. (This insures a load at main storage address zero.)

   b. Position CONSOLE MODE SELECT rotary switch at MS-WR.

   c. Position LOAD SELECT switch to the up position (either ALTERNATE or CR).

   d. Place the UTSSTD object deck in the card reader and press card reader START.

   e. Press RESET/LOAD.

*4. Set processor 4 P-register to zero.

   a. Position CONSOLE MODE SELECT switch to RF-WR.

   b. Position CONSOLE CONTROL SELECT switch to STOP/STEP.

   c. Position CONSOLE ADDRESS REGISTER SELECT switch to ADDRESS.

   d. Position CONSOLE DATA REGISTER SELECT switch to DATA.

   e. Press SYSTEM RESET.

*UTSSTD can be restarted at this point. It is not necessary to reload.

f. Press the CONSOLE ADDRESS REGISTER DISPLAY pushbutton/indicators to enter 00089 in the CONSOLE ADDRESS REGISTER.

g. Press the CONSOLE RUN pushbutton.

h. Press SYSTEM RESET.

5. Set the PROCESSOR SELECT rotary switch to 4.

6. Press PROCESSOR RUN.

The console responds with the header message TAPE-TO-DISC RESTORE and begins the series of parameter requests. See UTSSTD PARAMETERS. After each request, the operator enters the appropriate parameter and a carriage return or ETX.

If the utility detects an EOF on tape, the RESTORE ANOTHER PACK? message is sent. An EOV effects tape rewinding and unloading. The following message is sent to the operator:

NEXT BACKUP TAPE REQUIRED.

The FROM TAPE parameter request is then reissued to ask for the address and volume identifier of the next back-up tape for the disc. When restoration is completed, the following message is sent:

TAPE-TO-DISC RESTORE COMPLETE.

UTSSTD PROGRAM STOPS

The following program stop numbers can appear in the processor 4 P-register. When the error has been corrected the program must be restarted at address zero (step 4 of operating procedures).

Stop Number — Meaning of Stop

0002 — A branch to this address occurs after display of the message:

TAPE ERROR. . .
RETRY?

followed by an operator reply of N.

An automatic backspace and retry of the read sequence has occurred 5 times before UTSSTD displays this message. Cause for this persistent failure may be a defective spot on the tape or a defective tape drive. If possible, correct the error and restart the utility.

0004 — Modem or disconnect error for the operator's console. Correct the error and restart the utility.

Stop Number — Meaning of Stop

0006 — A branch to this address occurs when the user replies N to the message:

```
{TAPE} UNSOLICITED ATTENTION
{DISC}
CONTINUE?
```

or to the message BYPASS TRACK? If no error has actually occurred, restart the utility.

0008 — DEVICE cuu TIMED OUT is the message which precedes a branch to this address.

During the actual restore, this message and program stop indicate a hardware malfunction; specifically, channel-end is not being returned within a specified amount of time. When this situation occurs, no other message appears before the program stop.

During parameter solicitation, however, the message RETRY? follows the time-out message. The branch occurs only if the user has replied N.

The actual time-out is a time-out of the SIO instruction. This may occur during a normal I/O attempt or during the polling sequence. Recommended action: Take a dump and forward it and the console log to your Field Service Representative.

000A thru 000E — Stops at these addresses, should not occur.

Recommended action: Take a memory dump (preserving the micro-registers). Forward it and the console log to your Field Service Representative.

0010 — Irrecoverable disc search failure persists after 10 retries. Either the disc is in error or the drive is malfunctioning. If the error does not clear up after restarting the utility, reinitialize the disc. If the error persists on the reinitialized pack, dump memory, and forward the dump and console log to the Field Service Representative.

E. ERROR MESSAGES

This section lists the messages issued by the following utility programs. (There are no error messages for the Stand-Alone Memory Dump.)

Stand-Alone Disc-to-Tape Dump

Stand-Alone Tape-to-Disc Restore

Disc File Conversion Unit Copy

IBM Disc Volume to Memorex Disc Volume

Memorex Disc Volume to IBM Disc Volume

IBM Tape File to Memorex Tape File

Emulated Disc File to Memorex Disc File

File to File

Stand-Alone Disc Initialize

Rebuild Track

Tape Initialize

Change Volume Serial Number

System Catalog Display

Purge File

Allocate File

Stand-Alone Disc-to-Disc Copy

Catalog and Uncatalog

Load Universal Character Set

The utility program messages, except for the messages from the stand-alone utilities are preceded by an 8-character error code that has the following format:

PP	SS	EEE	T

where:

PP* is UT meaning utility program error.

SS* is CV for conversion utilities: UTCVIM, UTCVMI, and UTCVTM; is EM for the conversion utility UTCVEM; is FF for the file-to-file utility; and is SS for the systems service utilities.

EEE is a three-digit error number specifying the error within the utility program.

T is a one-digit number specifying the type of error where:

0 = no error
2 = warning
4 = error – may continue processing
8 = disaster

The stand-alone utilities send only the message text to the operator console.

Utility program messages reference Tables E-1 through E-5. Some messages also refer to I/O Control messages, Data Management messages, and System Loader messages in the **MRX/OS Messages** manual.

Table E-1. IFA Attention and P.D.A. Status Codes

YYYY	Bit	Definition
		Attention on Drive:
800y	0	0
400y	1	1
200y	2	2
100y	3	3
080y	4	4
040y	5	5
020y	6	6
010y	7	7
008y	8	Spare drive
004y	9	Indicates 100 cylinder IFA
	10-11	Unused
000y	12-15	Physical drive number (range 1-9)

*PPSS is DFCU for the UTDFCU utility program.

Table E-2. IFA Hardware Status Codes

xxxx	Bit	Status Definition
8000	0	IFA status not valid or command early
4000	1	IFA missed window or command early
2000	2	IFA window
1000	3	IFA track boundary
0800	4	IFA read/write termination
0400	5	IFA burst check error
0200	6	IFA lost data

*Detection of this bit by itself or in conjunction with bit 4 after a search command indicates a search-no-find condition. Detection of this bit by itself after issuing any command other than a search indicates disc write current sense.

Table E-3. Operation Codes for Disc and Tape I/O

Hex Code	Operation
01	Write
02	Read
	Codes exclusive to Disc I/O (DIO Usage)
04	Read without data transfer
08	Search
10	Format write
	Codes exclusive to Tape I/O (SIO Usage)
00	Test I/O
04	Sense
07	Rewind
0F	Rewind and unload
17	Erase gap
1F	Write tape mark
27	Backspace block
2F	Backspace file
37	Forward space block
3F	Forward space file
1B	Request track in error
0B	Set force error mode
C3	Clear force error mode

Table E-4. Sense Words for MRX 3237 Tape Drives

Bit	Designation		
	In Word 0	In Word 1	In Word 2
0	Command Reject	LRCR error	EP7
1	Intervention Required	VRC error	CRCP
2	Bus-out Check	CRCR error	CRC0
3	Equipment Check	Delay counter error	CRC1
4	Data Check	Write counter error	CRC2
5	Overrun	Read counter error	CRC3
6	Word count zero	R/W reg error	CRC4
7	Not capable	Noise	CRC5
8	Ready	EPP	CRC6
9	Not present	EP0	CRC7
10	BOT	EP1	Force error mode
11	EOT	EP2	PTIE*
12	Phase	EP3	PRAM ERROR*
13	File protect	EP4	POAM ERROR*
14	Tape mark	EP5	Reserved
15	MTIE	EP6	Reserved

*These conditions apply only to phase-encoded tape.

Table E-5. IOC Hardware Status Codes

yyyy	Bit	Unit Status Information
8000	0	Attention
4000	1	Status modifier
2000	2	Control unit end
1000	3	Busy
0800	4	Channel end
0400	5	Device end
0200	6	Unit check
0100	7	Unit exception

yyyy	Bit	Channel Status Information
0080	8	Initial selection sequence error
0040	9	Invalid command word
0020	10	Channel address/status check
0010	11	No 'REQUEST IN'
0008	12	Control check
0004	13	Transmission check
0002	14	Short buffer
0001	15	Reserved

The messages in this section are organized by utility name first and then message type, such as console message, SYSOUT file message, etc.

STAND-ALONE DISC-TO-TAPE DUMP MESSAGES

The normal termination message and the set of parameter request messages are included in under *Stand-Alone Disc-to-Tape Dump* in Section 2. The error messages are listed below.

CONSOLE MESSAGES

GENERAL INFORMATION MESSAGES

No reply is required for the following messages:

DISC-TO-TAPE DUMP mm-dd-yy

This is the program header message which precedes the parameter request sequence.

INVALID REPLY

This message is sent when the format or content of a reply to a message is invalid. This message is followed by a repeat of the original message.

REPLY CANCELLED

This message is sent when the reply to a message is terminated by a CAN, ENQ, or NAK. The original message is then repeated.

UPPER LIMIT < LOWER

During the parameter request sequence, the partial dump option has been selected and the address supplied by the user for the upper cylinder and head is less than that supplied for the lower. This is followed by a repeat of the parameter requests for the lower and upper limits.

MESSAGES SHARED BY DISC AND TAPE ERROR ROUTINES

$\begin{Bmatrix}\text{DISC}\\ \text{TAPE}\end{Bmatrix}$ SPECIFIED VOLUME ID UNEQUAL ACTUAL, vvvvvv.
CONTINUE?

> This message indicates that the disc or tape volume identifier, vvvvvv, is not equal to that specified in the reply to the parameter request. If the actual identifier contains an illegal character, a question mark is substituted for the character.
>
> If the operator replies Y or YES, the utility requests further parameters.
>
> If the operator replies N or defaults the reply, the utility repeats the original disc or tape parameter request.

$\begin{Bmatrix}\text{DISC}\\ \text{TAPE}\end{Bmatrix}$ UNSOLICITED ATTENTION
CONTINUE?

> An unsolicited attention occurred on the current disc or tape drive. This can occur at program load time or when a drive which has been offline is now ready or when the operator changes the mounting status. Such cases are normal. Unsolicited attentions occurring during the middle of the dump are probably abnormal.
>
> Reply Y to continue execution.
>
> A default reply or a reply of N causes execution to stop. Processor 4 will be at address 0006.

TAPE ERROR MESSAGES

In the following messages cuu contains the tape device address:

DEVICE cuu MOUNT ERROR
RETRY?

> This message occurs during the tape parameter request sequence and indicates that the WRITE ENABLE ring is missing from the tape on device cuu. The condition should be corrected before replying to the message. A reply of Y will cause the utility to rewind the tape and retry the I/O. A reply of N or a default reply causes the utility to resubmit the tape parameter request.

DEVICE cuu NOT A TAPE

> This message, occurring during the parameter request sequence, is followed by a repeat of the tape parameter request.

DEVICE cuu TIMED OUT

When this message occurs by itself, a hardware malfunction has occurred. Specifically, channel-end has not been returned within a specified amount of time. The utility branches to address 0008. See *UTSSDT Program Stops* in Section 2.

DEVICE cuu TIMED OUT
RETRY?

This message occurs during the parameter request sequence when the first SENSE command is issued for device cuu. The SENSE is retried if the user replies Y to this message. If the reply is N or defaulted, the utility branches to programmed stop address 0008. Consistent reappearance indicates a definite hardware malfunction. Reply 'N', take a storage dump and call the Field Service Representative.

DEVICE cuu TIMED OUT
STOP THE TAPE AND MOUNT ANOTHER! LABEL IS MISSING!

Stop the tape on device cuu immediately. This message occurs when a degaussed tape is mounted; no label is present. Operator intervention is the only way to stop the action. This message is followed by 'TO TAPE (cuu,volume)=', giving the user a chance to mount another tape for output.

DEFECTIVE TAPE VOLUME LABEL
RETRY?

An attempt to read the tape volume label has resulted in a length error detection of a tape mark or the first 4 characters do not equal VOL1. In all except the tape mark error, an automatic retry has been attempted 3 times before the message is displayed. A reply of Y causes the utility to rewind the tape and retry the read. If the reply is N or defaulted, the utility resubmits the parameter request.

TAPE ERROR-STATUS xxxx OP nn SENSE zzzz zzzz zzzz
RETRY?

This message appears when an I/O error has occurred for tape.

xxxx	contains the IOC Hardware Status (see Table E-5)
nn	contains the hexadecimal code for the tape operation being performed (see Table E-3).
zzzz, etc.	contain the sense information returned from tape (see Table E-4).

If the operator replies Y to the message, the utility will reposition the tape and retry the operation that failed. (Unsuccessful repositioning causes the utility to branch to address 0002. See *UTSSDT Program Stops* in Section 2.)

Utility action if the reply is N or defaulted depends upon the phase of execution. If during the tape parameter request routine, the message 'TO TAPE (cuu,volume)=' will be re-sent. Otherwise, the utility will halt at address 0002. (See *UTSSDT Program Stops* in Section 2.)

Messages Relating to Defective/Alternate Tracks

All messages in this section are informative. That is, they simply report automatic non-standard action that is being taken by the utility. In these messages, ccc and hh contain the cylinder and head numbers, in decimal, of the associated disc pack track.

For those conditions whose final message line is TRACK WILL BE BYPASSED, utility action proceeds to the end-of-track routine. This routine tests to see if more tracks are to be dumped, proceeding accordingly to the next track or going through the post-dump parameter requests. See *UTSSDT Parameters* in Section 2.

CYL ccc HD hh IS AN ALT TRK
TRACK WILL BE BYPASSED

A track within the dump limits which is supposed to be a primary was found to be an assigned alternate.

CYL ccc HD hh IS DEFECTIVE
ALT CYL ccc HD hh WILL BE TRIED

A primary track within the dump limits is defective and the dump will be made from the assigned alternate.

CYL ccc HD hh IS DEFECTIVE
NO ALT ALLOWED
TRACK WILL BE BYPASSED

It is illegal to assign an alternate to a defective primary when the primary is cylinder 0, head 0 or in the range of control storage tracks.

CYL ccc HD hh IS DEFECTIVE
NO ALT ASSIGNED
TRACK WILL BE BYPASSED

The defective primary track has no assigned alternate. The track will be automatically bypassed.

CYL ccc HD hh IS DEFECTIVE
TRACK WILL BE BYPASSED

This occurs only when an assigned alternate is defective. The message indicating the primary-defective track and the alternate to be tried precedes this message.

CYL ccc HD hh IS PRIMARY!
TRACK WILL BE BYPASSED

This appears when the flag byte of a track which is assigned as an alternate from a primary indicates that the track is really not assigned; that is, the assigned alternate is labeled primary. The message showing the original defective-primary and the alternate to be tried precedes this message.

Recoverable Disc Errors

For the following disc error messages, all of which include RETRY?, utility action is as follows:

> If the operator replies Y, the utility retries the operation or command program that failed. Action for an N or default reply depends on the programming phase – if during the parameter setup sequence, the utility resubmits the parameter request message. During the actual dump phase, the track will be bypassed, a message to that effect will be sent to the console, and the utility will proceed to the next track.

DISC ERROR-STATUS xxxx OP nn CYL ccc HD hh text
RETRY?

An I/O error has occurred on the disc.

xxxx contains the IFA Hardware Status. See Table E-2.

nn Contains the hexadecimal code of the operation which failed. (See Table E-3.)

ccc and hh contain (in decimal) the cylinder and head numbers, respectively, of the track on which the error occurred.

text describes the field on the track where the error occurred and will be one of the following:

HOME ADDRESS
C.S. DATA
REC rrr COUNT
REC rrr DATA
REC rrr EMU
REC rrr KEY

rrr is the number of the record where the I/O failed. An exception to this occurs when the failing command word is a READ for a RECORD N COUNT (N > 0). In this case rrr is set equal to '?'.

DISC ERROR-STATUS xxxx WHILE SEEKING CYL ccc HD hh
RETRY?

A possibly recoverable error occurred during the seek routines for the disc.

xxxx is the IFA Hardware Status (see Table E-2).

ccc and hh are the cylinder and head numbers of the track which the routine was trying to seek.

DISC ERROR-STATUS xxxx WHILE SELECTING DRIVE n
RETRY?

An error has occurred during the disc drive selection sequence.

xxxx is the IFA Hardware Status (see Table E-2).

n is the disc drive number.

DISC MOUNT ERROR
RETRY?

The disc is mounted READ/WRITE instead of READ-ONLY. If the drive number is correct, the operator should change the mount status before replying to the message.*

DISC OFFLINE
RETRY?

The disc is not yet on line. If the drive number is correct, the user should wait for it to be fully powered up before replying to the message.* If the drive is fully powered up, the ENABLE/DISABLE switch should be set to ENABLE.

Irrecoverable Disc Error

DISC ERROR-STATUS xxxx WHILE SEEKING CYL ccc HD hh
BUT ATTN=yyyy. . .FATAL ERROR

This message indicates a disc hardware malfunction. The sequence of events has been (1) a seek was performed against a cylinder and head, (2) after the seek, when the disc was no longer 'busy', (3) an attention failed to come up for the drive. After this message is displayed, the utility turns itself off.

xxxx	is the IFA Hardware Status (see Table E-2).
ccc and hh	are, in decimal, the cylinder and head numbers of the track on which the failure occurred.
yyyy	is the IFA Attention and P.D.A. Status (see Table E-1).

Recommended action: Take a storage dump and forward it with the console log to your Field Service Representative.

STAND-ALONE TAPE-TO-DISC RESTORE MESSAGES

The normal termination message and the set of parameter request messages are included under *Stand-Alone Tape-to-Disc Restore* in Section 2. The error messages are listed below.

CONSOLE MESSAGES

GENERAL INFORMATION MESSAGES

No reply is required.

*Change of mount condition or power up on a drive will generate the message

'DISC-UNSOLICITED ATTENTION
CONTINUE?'

The operator should reply Y because these are typical normal causes of the unsolicited attention message.

INVALID REPLY

This message is sent when the format or content of a reply to a message is invalid. This message is followed by a repeat of the original message.

REPLY CANCELLED

This message is sent when the reply to a message is terminated by a CAN, ENQ, or NAK. The original message is then repeated.

TAPE-TO-DISC RESTORE mm-dd-yy

This program header message precedes the parameter request sequence. The rest of the normal parameter request messages are described under *UTSSTD Parameters* in Section 2.

MESSAGES SHARED BY DISC AND TAPE ERROR ROUTINES

$\left\{\begin{array}{l}\text{DISC}\\ \text{TAPE}\end{array}\right\}$ SPECIFIED VOLUME ID UNEQUAL ACTUAL, vvvvvv.
CONTINUE?

This message indicates that the volume identifier vvvvvv, on the device is not equal to that specified in the reply to the parameter request. If the actual identifier contains an illegal character, a question mark is substituted for the character.

If the operator replies Y or YES, the utility requests further parameters. If the reply is N or defaulted, the utility repeats the original parameter request.

$\left\{\begin{array}{l}\text{DISC}\\ \text{TAPE}\end{array}\right\}$ UNSOLICITED ATTENTION
CONTINUE?

An unsolicited attention occurred on the current disc or drive. This can occur at program load time or when a drive which has been offline is now ready or when the operator changes the mounting status. Such cases are normal. Unsolicited attentions occurring during the middle of the dump are probably abnormal.

Reply Y to continue execution.

A default reply or a reply of N causes execution to stop at programmed address 0006 for processor 4. See *UTSSDT Program Stops* in Section 2.

TAPE ERROR MESSAGES

BACKUP VOLID UNEQUAL DISC VOLID
CONTINUE?

The volume identifier of the disc dumped to this tape is not the same as that of the disc to which the restore is to be made. If the reply is Y the utility proceeds to the restore. If the reply is N or defaulted, the utility resends the message FROM TAPE (cuu,volume)=.

DEFECTIVE TAPE VOLUME LABEL
RETRY?

An attempt to read the tape volume label has resulted in a length error, detection of a tape mark, or the first 4 characters do not equal VOL1. In all except the tape mark error, an automatic retry has been attempted three times before the message is displayed. A reply of Y causes the utility to rewind the tape and retry the read. If the reply is N or defaulted, the utility resubmits the tape.

DEVICE cuu MOUNT ERROR
RETRY?

This message occurs during the tape parameter request sequence. It indicates that the WRITE ENABLE ring must be removed from the tape. Correct the condition before replying to the message. A reply of Y will cause the utility to retry the operation which failed. A reply of N or the default is followed by a repeat of the tape parameter request message.

DEVICE cuu NOT A TAPE

This message, occurring during the tape parameter request sequence, is followed by a repeat of the tape parameter request.

DEVICE cuu TIMED OUT

This message, when occurring by itself, indicates a hardware malfunction. After sending this message the utility branches to programmed stop address 0008. (See *UTSSTD Program Stops* in Section 2.)

DEVICE cuu TIMED OUT
RETRY?

This message occurs during the parameter request sequence. A hardware malfunction may have occurred; however, the user may retry the SENSE command which failed by replying Y to the message. A reply of N or a default reply causes the utility to branch to programmed stop address 0008.

Consistent reappearance of this message definitely indicates a hardware malfunction. Reply N, request a storage dump, and forward the dump and the console log to your Field Service Representative.

DEVICE cuu TIMED OUT
STOP THE TAPE AND MOUNT ANOTHER!

Stop the tape. This message occurs if the tape reading fails at some point after the device label because the rest of the tape is blank. Operator intervention is required to stop the tape motion. A different tape will be needed. The utility next sends the tape a parameter request message to give the operator a chance to restore from a different tape.

DEVICE cuu TIMED OUT
STOP THE TAPE AND MOUNT ANOTHER! LABEL IS MISSING!

Stop the tape on device cuu immediately. This message occurs when a degaussed tape is mounted; no label is present. Operator intervention is the only way to stop the action. This message is followed by the tape parameter request message.

TAPE ERROR-STATUS xxxx OP nn SENSE zzzz zzzz zzzz
RETRY?

This message appears when an I/O error has occurred for tape

xxxx contains the IOC Hardware Status (see Table E-5).

nn contains the hexadecimal code for the tape operation being performed (see Table E-3).

zzzz, etc. contain the sense information returned from tape (see Table E-4).

If the operator replies Y to the message, the utility will reposition the tape and retry the operation that failed. Unsuccessful repositioning causes the utility to branch to halt address 0002.

Utility action if the reply is 'N' or defaulted depends upon the phase of execution. If during the tape parameter request routine, the parameter request will be re-sent. Otherwise, the utility will halt at address 0002.

TAPE FORMAT ERROR. LAST BLOCK NUMBER=nnnn.
Occurs if a block on the tape is not one of the following:

header label
trailer label
standard block created by UTSSDT

or if no tape mark appears after the second header label.

nnnn=0	if the error occurs in the header label -- tape mark region.
nnnn	otherwise, is the last block successfully read. The utility rewinds and unloads the tape and then issues a parameter request message.

TAPE IS NOT A BACKUP
The first header label indicates that the tape mounted was not created by UTSSDT and thus does not meet the specifications required by this utility. The tape is rewound and unloaded and the tape parameter request message is repeated.

TAPE VOLUME SEQUENCE ERROR
CONTINUE?
This message is sent if the tapes are out of sequence. A reply of Y will cause the utility to proceed to the restore. A reply of N or a default reply proceeds to rewind and unload the tape, send the message NEW TAPE REQUIRED and then repeat the tape parameter request message.

DISC ERROR MESSAGES

Messages Related to Defective/Alternate Tracks

The following messages are informative; they report action by the utility program.

CYL ccc HD hh IS AN ALT TRK
TRACK WILL BE BYPASSED
The address of the primary track originally dumped to tape corresponds to the address of an assigned alternate on the disc.

CYL ccc HD hh IS DEFECTIVE
ALT CYL ccc HD hh WILL BE TRIED
The track to be restored is defective; restoration will be attempted on the assigned alternate.

CYL ccc HD hh IS DEFECTIVE
NO ALT ALLOWED
TRACK WILL BE BYPASSED
If track 0 or a Control Storage track, it is illegal to assign an alternate to a defective primary.

```
CYL ccc HD hh IS DEFECTIVE
NO ALT ASSIGNED
TRACK WILL BE BYPASSED
```

The track to be restored is defective and has no assigned alternate.

```
CYL ccc HD hh IS DEFECTIVE
TRACK WILL BE BYPASSED
```

This message follows the message indicating that an alternate track is being tried; it indicates that the assigned alternate is itself defective.

```
CYL ccc HD hh IS A PRIMARY!
TRACK WILL BE BYPASSED
```

This message follows the message indicating that an alternate track is being tried; it shows that the assumed alternate track is labeled as primary.

Recoverable Disc Errors

If the operator replies Y, the utility retries the operation or command program that failed. Action for an N or default reply depends on the programming phase. During the parameter setup sequence, the utility resubmits the parameter request message. During the actual restore, however, the message BYPASS TRACK? is issued. A reply of Y to this message produces the message TRACK WILL BE BYPASSED followed by normal processing of the next tape block. If the BYPASS TRACK message is answered N or defaulted, control branches to programmed stop address 0006.

```
DISC ERROR-STATUS xxxx OP nn CYL ccc HD hh text
RETRY?
```

An I/O error has occurred on the disc.

xxxx contains the IFA Hardware Status (see Table E-2).

nn contains the hexadecimal code of the operation that failed (see Table E-3).

ccc and hh are the decimal representation of the cylinder and head numbers of the track on which the error occurred.

text will describe the field on the track where the error occurred and will be one of the following:

```
HOME ADDRESS
C.S. DATA
REC rrr COUNT
REC rrr DATA
REC rrr EMU
REC rrr KEY
```

rrr is the number of the record where the I/O failed. An exception to this occurs when the failing command word is a READ for a RECORD N COUNT (N > 0). In this case rrr is set equal to '?'.

DISC ERROR-STATUS xxxx WHILE SEEKING CYL ccc HD hh
RETRY?

A possibly recoverable error occurred during the seek routines for the disc.

xxxx	is the IFA Hardware Status (see Table E-2).
ccc and hh	are the cylinder and head numbers of the track which the routine was trying to seek.

If the operator replies Y, the utility retries the operation or command program that failed. Action for an N or default reply depends on the programming phase. During the parameter setup sequence, the utility resubmits the parameter request message. During the actual restore, however, the message BYPASS TRACK? is issued. A reply of Y to this message produces the message TRACK WILL BE BYPASSED followed by normal processing of the next tape block. If the BYPASS TRACK message is answered N or defaulted, control branches to programmed stop address 0006.

DISC ERROR-STATUS xxxx WHILE SELECTING DRIVE n
RETRY?

An error has occurred during the disc drive selection sequence.

xxxx	is the IFA Hardware Status (see Table E-2).
n	is the disc drive number.

If the operator replies Y, the utility retries the operation or command program that failed. Action for an N or default reply depends on the programming phase. During the parameter setup sequence, the utility resubmits the parameter request message. During the actual restore, however, the message BYPASS TRACK? is issued. A reply of Y to this message produces the message TRACK WILL BE BYPASSED followed by normal processing of the next tape block. If the BYPASS TRACK message is answered N or defaulted, control branches to programmed stop address 0006.

DISC MOUNT ERROR
RETRY?

The disc is mounted READ-ONLY instead of READ-WRITE. If the drive number is correct, the operator should change the mount status before replying to the message.

If the operator replies Y, the utility retries the operation or command program that failed. Action for an N or default reply depends on the programming phase. During the parameter setup sequence, the utility resubmits the parameter request message. During the actual restore, however, the message BYPASS TRACK? is issued. A reply of Y to this message produces the message TRACK WILL BE BYPASSED followed by normal processing of the next tape block. If the BYPASS TRACK message is answered N or defaulted, control branches to programmed stop address 0006.

An unsolicited attention will be generated when the mount status is changed. This is a normal condition and the operator should continue.

DISC OFFLINE
RETRY?

The disc is not yet on line. If the drive number is correct, the user should wait for it to be fully powered up before replying to the message. If the drive is fully powered up, the ENABLE/DISABLE switch should be set to ENABLE.

If the operator replies Y, the utility retries the operation or command program that failed. Action for an N or default reply depends on the programming phase. During the parameter setup sequence, the utility resubmits the parameter request message. During the actual restore, however, the message BYPASS TRACK? is issued. A reply of Y to this message produces the message TRACK WILL BE BYPASSED followed by normal processing of the next tape block. If the BYPASS TRACK message is answered N or defaulted, control branches to programmed stop address 0006.

Occurrence of this message during the dump phase is extremely unlikely and would indicate serious operator error or hardware malfunction.

Changes of the power up condition will generate an unsolicited attention message. The operator should reply Y to the attention message.

Irrecoverable Disc Error

DISC ERROR-STATUS xxxx WHILE SEEKING CYL ccc HD hh
BUT ATTN=yyyy. . .FATAL ERROR

This message indicates a disc hardware malfunction. The sequence of events has been (1) a seek was performed against a cylinder and head, (2) after the seek, when the disc was no longer 'busy', (3) an attention failed to come up for the drive. After this message is displayed, the utility turns itself off.

xxxx	is the IFA Hardware Status (see Table E-2).
yyyy	is the IFA Attention and P.D.A. Status (see Table E-1).
cc and hh	are the cylinder and head numbers (in decimal) of the track for which the failure occurred.

DISC FILE CONVERSION UNIT COPY (UTDFCU) MESSAGES

CONSOLE AND SYSOUT MESSAGES

All UTDFCU messages appear both at the operator's console and on the SYSOUT file.

ERROR CODE	MESSAGE TEXT
DFCU0010	DFCU READY – ENTER COMMAND The program is ready to accept a test command or the COPY command.
DFCU0020	DFCU READY FOR NEXT COMMAND The program is ready to accept a command. If the previous command routine has not been completed, the response to this message will override the previous command.

ERROR CODE	MESSAGE TEXT
DFCU0030	INVALID COMMAND – TRY AGAIN The last command entered was found to be in error. Loop up the correct format and re-enter the command.
DFCU0040	DFCU TEST FINISHED The test routine for the last command has been completed. Enter the next command as the response for message DFCU0020.
DFCU0050	command STATUS (xxxx) ON CYL cc HEAD h RECORD r An error has occurred on disc address cchr (hexadecimal) xxxx the IFA hardware status (see Table E-2) command the I/O operation which caused the error
DFCU0060	DATA ERR aaaaaaaaaa S/B bbbbbbbbbb ON CYL cc HEAD h This message may occur during the VFYSEC or FVYTRK tests. The actual data read from the disc appears as a. . .a;the test pattern from the special test pack appears as b. . .b. This message will appear each time an error is detected as the data buffer is scanned. If typeouts become excessive, they may be bypassed with the SKIPTO command.
DFCU0070	HOME ADDR aaaaaaaa S/B bbbbbbbb ON CYL cc HEAD h The home address a-a which was read is incorrect.
DFCU0080	DFCU FILE UNSAFE, RESET IT AND RESPOND WITH (ETX) A file unsafe condition has occurred on either the DFCU, the drive, or both. If the drive is unsafe it must be powered down, then back up to remove the condition. To remove unsafe from the DFCU operate the MASTER CLEAR switch on the DFCU maintenance panel. When the condition has been reset, press INT, enter the message number and one space, press ETX to resume processing.
DFCU0090	IFA TEST PATTERN pppppppp This message is displayed when in IFATEST mode of operation each time that the data switches on the DFCU maintenance panel are changed.
DFCU0100	FILE filename NOT IN FDT STRING This file cannot be found in the FDT string. Should never occur if the file has been opened properly.
DFCU0110	FILE filename ALLOCATED ON NON-CONTIGUOUS SPACE Correct your Control Language and resubmit the job.
DFCU0120	FILE filename DOES NOT START ON A CYLINDER BOUNDARY Correct your Control Language and resubmit the job.
DFCU0130	FILE filename HAS INVALID SIZE File size not 500, 520, 1000, or 1020 blocks. Correct your Control Language to properly allocate the file and resubmit the job.
DFCU0140	COPYING nnnn TRACKS FROM 1316 VOL volsn
DFCU0141	TO MRX FILE filename ON VOL volid These messages describe the file which is being copied. They are typed out as the copy begins.

ERROR CODE	MESSAGE TEXT
DFCU0150	VOLUME xxxxxx ON DFCU – MOUNT yyyyyy The input label check failed for the 1316 pack which is mounted on the DFCU. xxxxxx is the volume serial number of the pack which is mounted; yyyyyy is the volume serial number of the pack which should be mounted. Mount the correct pack on the DFCU and reply with Y or N.
DFCU0160	COPY COMPLETED This message appears upon successful completion of the copy.
DFCU0170	RETRY? ANSWER Y OR N This message refers to the error message just typed out. The Operation has already been tried 10 times. Enter Y to try again; N to abort the run.
DFCU0180	COPY ABORTED This message appears when run has been discontinued prior to completion. The previous error message describes the reason for premature termination.
DFCU0190	DEFECTIVE TRACK FLAGGED ON CYL cc HEAD h A defective track has been encountered while copying to an EMULATOR file which has been initialized with the SUPALT option (26 or 51 cylinders). During this mode of copy the input pack must contain no defective tracks, as cylinders 1 through 3 are assumed to contain data and not be used as alternate track areas.
DFCU0200	CANNOT RECOVER DEF TRACK ON CYL cc HEAD h On IBM 1316 tracks which are flagged as 'defective', the assigned alternate track address is recorded in each of the 10 count fields on the defective track. At least one of these count fields must be readable or the pack is unusable. This pack is unusable.
DFCU0210	INVALID FILENAME filename1 JCL SPECIFIES filename2 Where filename1 is from the 'COPY' command and filename2 is from the //DEF card for EMUFILE. This is to protect the user from inadvertently writing over an existing file.

IBM DISC VOLUME TO MEMOREX DISC VOLUME (UTCVIM) MESSAGES

Messages UTCV101x to UTCV1118 appear on SYSOUT following the //PAR statement containing the error(s). Error UTCV1068 applies to the previous //PAR statement if it did not terminate with a comma.

SYSOUT FILE ERROR MESSAGES

ERROR CODE	MESSAGE TEXT
UTCV100x	END OF IBM TO MEMOREX CONVERSION. This message is output upon normal completion of UTCVIM. The variable x contains the highest return code encountered.
UTCV1018	PARAMETER CARD FORMAT ERROR – ONLY ONE COMMA MAY SEPARATE A PARAMETER FROM A KEYWORD. More than one comma separated a parameter from a keyword.
UTCV1028	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST BE DELIMITED BY AN EQUAL SIGN. While scanning for the end of a keyword, a space was found.
UTCV1038	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST PRECEDE AN EQUAL SIGN. An equal sign was found without a keyword preceding it.
UTCV1048	INVALID KEYWORD xxx. . . . The keyword xxx. . . . is invalid.
UTCV1058	DUPLICATE KEYWORD xxxxxxxx. Two keywords xxxxxxxx were entered via //PAR statements.
UTCV1068	A COMMA MUST SEPARATE PARAMETERS FROM KEYWORDS. A comma did not separate a parameter from a keyword.
UTCV1078	INVALID PARAMETER FOLLOWING KEYWORD xxxxxxxx. The parameter following the keyword xxxxxxxx contains an illegal character or its length is incorrect.
UTCV1088	TOO MANY SUBPARAMETERS ENTERED FOR KEYWORD xxxxxxxx. The count of subparameters following keyword xxxxxxxx is greater than the maximum allowed.
UTCV1098	CONTINUATION WAS INDICATED BUT NO CONTINUATION CARD WAS FOUND. The last //PAR statement ended with a comma.
UTCV1108	A LEFT PARENTHESIS WAS FOUND WITHOUT A RIGHT PARENTHESIS. Incorrect termination of subparameterization.
UTCV1118	REQUIRED KEYWORD xxxxxxxx WAS NOT FOUND. A required keyword was not entered through a //PAR statement.
UTCV1134	POSSIBLE VTOC ERROR EXISTS – TO CONTINUE ANSWER WITH Y, TO ABORT ANSWER WITH ETX. This message appears on the console when a VTOC label of an OS pack indicates a DADSM function terminated prematurely. Conversion may be attempted but file space may not be accurately described.

ERROR CODE	MESSAGE TEXT
UTCV1144	POSSIBLE VTOC ERRORS EXIST { JOB ABORTED BY OPERATOR JOB CONTINUED BY OPERATOR } This message appears on SYSOUT and indicates the operator's response to error UTCV1134.
UTCV1158	IRRECOVERABLE DISC I/O ERROR WHEN x. . .x CYLINDER yyy HEAD zz RETURN INFORMATION = xxnn RETURN HARDWARE STATUS = xxxx. (See Table E-2.) I/O operation x. . .x failed to complete successfully on cylinder yyy, head zz. Refer to the I/O Control message where nn are the two rightmost hex status completion code digits.
UTCV1168	VOLUME ID DOES NOT MATCH VOLUME ID ENTERED VIA THE VOL KEYWORD. Incorrect volume was mounted or the parameter following the VOL keyword is incorrect.
UTCV1178	THE PACK CATALOG FILE WAS NOT FOUND IN THE VTOC USING FILENAME=x. . .x. Space for a pack catalog was not allocated prior to conversion or an incorrect parameter followed keyword PCNAME.
UTCV1188	A DISC DRIVE WAS NOT ASSIGNED TO THIS JOB. The unit table does not contain a disc drive assigned to this job.
UTCV1198	THE FIRST RECORD OF THE VTOC IS NOT A FORMAT 4 LABEL.
UTCV1208	VTOC FORMAT 4 LABEL INDICATED THERE WERE FORMAT 5 LABELS PRESENT BUT NO FORMAT 5 LABELS WERE FOUND.
UTCV1218	AN UNALLOCATED TRACK WAS NOT FOUND TO WRITE TRACK 0 ON. An empty track is not available for IBM's track 0; conversion cannot be completed.
UTCV1228	THE KEY LENGTH PLUS THE DATA LENGTH OF A TRACK 0 RECORD IS TOO LARGE FOR THE AVAILABLE BUFFER. The largest buffer available to UTCVIM will not contain the key and the data of a track 0 record.
UTCV1238	AFTER TRUNCATION OF FILE NAMES TO 17 CHARACTERS, TWO FILES EXIST WITH THE FILE NAME xxx. . . . Two files may not have the same file name. Conversion is terminated.
UTCV1248	THERE WAS NOT ENOUGH SPACE ALLOCATED FOR THE PACK CATALOG. The space allocated for the pack catalog was not large enough.
UTCV1258	THE SPACE FOR THE PACK CATALOG DOES NOT BEGIN ON A CYLINDER BOUNDARY.

MEMOREX DISC VOLUME TO IBM DISC VOLUME (UTCVMI) MESSAGES

Messages UTCV2018 to UTCV2118 appear on SYSOUT following the //PAR statement containing the error(s). Error UTCV2068 applies to the previous //PAR statement if it did not terminate with a comma.

SYSOUT FILE ERROR MESSAGES

ERROR CODE	MESSAGE TEXT
UTCV2000	END OF MEMOREX TO IBM CONVERSION. This message is output upon normal completion of UTCVMI.
UTCV2018	PARAMETER CARD FORMAT ERROR – ONLY ONE COMMA MAY SEPARATE A PARAMETER FROM A KEYWORD. More than one comma separated a parameter from a keyword.
UTCV2028	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST BE DELIMITED BY AN EQUAL SIGN. While scanning for the end of a keyword, a space was found.
UTCV2038	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST PRECEDE AN EQUAL SIGN. An equal sign was found without a keyword preceding it.
UTCV2048	INVALID KEYWORD x. . .x. The keyword x. . .x is invalid.
UTCV2058	DUPLICATE KEYWORD. Two keywords x. . .x were entered via //PAR statements.
UTCV2068	A COMMA MUST SEPARATE PARAMETERS FROM KEYWORDS. A comma did not separate a parameter from a keyword.
UTCV2078	INVALID PARAMETER FOLLOWING KEYWORD xxxxxxxx. The parameter following the keyword xxxxxxxx contains an illegal character or its length is incorrect.
UTCV2088	TOO MANY SUBPARAMETERS ENTERED FOR KEYWORD xxxxxxxx. The count of subparameters following the keyword xxxxxxxx is greater than the maximum allowed.
UTCV2098	CONTINUATION WAS INDICATED BUT NO CONTINUATION CARD WAS FOUND. The last //PAR statement ended with a comma.
UTCV2108	A LEFT PARENTHESIS WAS FOUND WITHOUT A RIGHT PARENTHESIS. Incorrect termination of subparameterization.
UTCV2118	REQUIRED KEYWORD xxxxxxxx WAS NOT FOUND. A required keyword was not entered through a //PAR statement.

ERROR CODE	MESSAGE TEXT
UTCV2158	IRRECOVERABLE DISC I/O ERROR WHEN xxx. . . .CYLINDER yyy HEAD zz RETURN INFORMATION = xxnn RETURN HARDWARE STATUS = xxxx. (See Table E-2.) An I/O operation xxx. . . .failed to complete successfully on cylinder yyy head zz. Refer to the I/O Control message where nn are the two rightmost hex status completion code digits.
UTCV2168	VOLUME ID DOES NOT MATCH VOLUME ID ENTERED VIA THE VOL KEYWORD. Incorrect volume was mounted or the parameter following the VOL keyword is incorrect.
UTCV2178	A DISC DRIVE WAS NOT ASSIGNED TO THIS JOB. This unit table does not contain a disc drive assigned to this job.
UTCV2188	THE FIRST RECORD OF THE VTOC IS NOT A FORMAT 4 LABEL.
UTCV2198	THE KEY LENGTH PLUS THE DATA LENGTH OF A TRACK 0 RECORD IS TOO LARGE FOR THE AVAILABLE BUFFER. The largest buffer available to UTCVMI will not contain the key and the data of a track 0 record.

IBM TAPE FILE TO MEMOREX FILE (UTCVTM) MESSAGES

Messages UTCV3018 to UTCV3118 appear on SYSOUT following the //PAR statement containing the error(s). Error UTCV3068 applies to the previous //PAR statement if it did not end with a comma.

SYSOUT FILE ERROR MESSAGES

ERROR CODE	MESSAGE TEXT
UTCV3000	UTCVTM NORMAL COMPLETION. Output upon normal completion of UTCVTM.
UTCV3018	PARAMETER CARD FORMAT ERROR – ONLY ONE COMMA MAY SEPARATE A PARAMETER FROM A KEYWORD. More than one comma separated a parameter from a keyword.
UTCV3028	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST BE DELIMITED BY AN EQUAL SIGN. While scanning for the end of a keyword, a space was found.
UTCV3038	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST PRECEDE AN EQUAL SIGN. An equal sign was found without a preceding keyword.

ERROR CODE	MESSAGE TEXT
UTCV3048	INVALID KEYWORD x. . .x. The keyword x. . .x is invalid.
UTCV3058	DUPLICATE KEYWORD x. . .x. Two identical keywords x. . , .x were entered via //PAR statements.
UTCV3068	A COMMA MUST SEPARATE PARAMETERS FROM KEYWORDS. A comma did not separate a parameter from a keyword.
UTCV3078	INVALID PARAMETER FOLLOWING KEYWORD x. . .x. The parameter following the keyword x. . .x contains an illegal character or its length is incorrect.
UTCV3098	CONTINUATION WAS INDICATED BUT NO CONTINUATION CARD WAS FOUND. The last //PAR statement ended with a comma.
UTCV3118	REQUIRED KEYWORD x. . .x WAS NOT FOUND. A required keyword was not entered through a //PAR statement.
UTCV3128	THE LARGEST POSSIBLE BLOCK IS TOO LARGE FOR THE AVAILABLE BUFFER SPACE. The largest buffer available to UTCVTM will not contain the largest possible block on the tape.

EMULATED IBM DISC FILE TO MEMOREX DISC FILE (UTCVEM) MESSAGES

The messages from this utility program appear on the operator console. The I preceding each message indicates that no operator reply is required.

CONSOLE MESSAGES

Only the message error code is printed on the console. The description does *not* follow the message.

ERROR CODE	MESSAGE DESCRIPTION
I UTEM0018	No M20BLK= parameter was found when one was required.
I UTEM0028 xxnn $\begin{Bmatrix}A\\O\end{Bmatrix}$	A logical I/O error has occurred. Refer to the Data Management GET/PUT message where nn are the two rightmost hex status completion code digits. $\begin{Bmatrix}A\\O\end{Bmatrix}$ indicates the file on which the error occurred, where A = alternate and O = output.

ERROR CODE	MESSAGE DESCRIPTION
I UTEM0038	The output file was not sequential or indexed.
I UTEM0048	The output file is indexed and the input (M20) file is sequential. KEYLEN= and KEYPOS= parameters were not specified.
I UTEM0058 nnnn	The size of the buffers (MRX) exceeds available space by nnnn bytes.
I UTEM0068 xxxxxx	This is a parameterization error where xxxxxx is the keyword of the invalid parameter.
I UTEM0078	The device assigned to an alternate file is not disc or tape.
I UTEM0088	The block size and/or the record size is invalid; the resulting blocking factor would be zero.

FILE TO FILE (UTFF) MESSAGES

The File to File utility program issues only SYSOUT file messages.

The error messages are preceded by an 8-character error code. The format is as follows:

UTFFxxxy

where:

UT identifies the error as a utility program error.

FF identifies the utility program as the File-To-File program.

xxx is a 3-digit error number specifying the error within the utility program.

y is a 1-digit number specifying the status of the error.

where 0 = no error
2 = warning
4 = error may continue processing
8 = disaster

SYSOUT FILE ERROR MESSAGES

ERROR CODE	MESSAGE TEXT
UTFF0000	FILE TO FILE UTILITY COMPLETED.
UTFF0018	KEYWORD PREVIOUSLY DEFINED x. . .x.
UTFF0028	INVALID VALUE FOR x. . .x.

ERROR CODE	MESSAGE TEXT
UTFF0038	IMBEDDED BLANK IN PARAMETER STATEMENT.
UTFF0048	INVALID KEYWORD x. . .x.
UTFF0058	VALUE OF x. . .x IS NOT ON ONE PARAMETER STATEMENT.
UTFF0068	'PAGHDR' VALUE IS NOT ENCLOSED WITH APOSTROPHES.
UTFF0078	'PAGHDR' IS GREATER THAN 132 CHARACTERS.
UTFF0088	INVALID DELIMITER FOLLOWING x. . .x.
UTFF0098	VALUE OF x. . .x IS NOT ENCLOSED WITH PARENTHESES.
UTFF0108	VALUE OF x. . .x IS NOT DELIMITED WITH A COMMA WITHIN THE PARENTHESES.
UTFF0118	MUTUALLY EXCLUSIVE KEYWORDS – 'LIST' AND 'DISPLAY'.
UTFF0128	INVALID INPUT FILE.
UTFF0138	INVALID OUTPUT FILE.
UTFF0148	ILLEGAL COMBINATION OF ACCESS METHODS.
UTFF0158	INPUT 'OPEN' ERROR.
UTFF0168	INPUT 'OPENL' ERROR.
UTFF0178	OUTPUT 'OPEN' ERROR.
UTFF0188	OUTPUT 'OPENL' ERROR.
UTFF0198	BINARY INPUT OR PUNCH OUTPUT.
UTFF0208	INPUT AND OUTPUT RECORD SIZE NOT EQUAL.
UTFF0218	ERROR DURING TYPE MACRO.
UTFF0228	ERROR DURING DISPLAY MACRO.
UTFF0238	ERROR DURING 'LOAD' OF PRINTER MODULE.
UTFF0248	INSUFFICIENT BUFFER SPACE.
UTFF0258	'READ' ERROR.
UTFF0268	INPUT 'CLOVE' ERROR.
UTFF0278	'WRITE' ERROR.
UTFF0288	OUTPUT 'CLOVE' ERROR.

ERROR CODE	MESSAGE TEXT
UTFF0298	EOA ON OUTPUT.
UTFF0308	'PUT' ERROR.
UTFF0318	'GET' ERROR.
UTFF0328	INPUT 'CLOSE' ERROR.
UTFF0338	INPUT 'CLOSEL' ERROR.
UTFF0348	OUTPUT 'CLOSE' ERROR.
UTFF0358	OUTPUT 'CLOSEL' ERROR.
UTFF0368	UNEQUAL BLOCK SIZES FOR RELATIVE INPUT FILE.
UTFF0378	INPUT AND OUTPUT KEYSIZES ARE NOT EQUAL.
UTFF0388	SUM OF RELATIVE KEY POSITION AND KEY SIZE EXCEEDS THE RECORD SIZE.
UTFF0398	ERROR DURING LABRTN MACRO.
UTFF0408	ERROR DURING ACCEPT MACRO.
UTFF0418	INSUFFICIENT OUTPUT INDEX BLOCKING FACTOR.

TAPE INITIALIZE

CONSOLE MESSAGES

Messages sent to the operator by the Tape Initialize Utility are listed below.

ERROR CODE **MESSAGE TEXT**

UTSS0180 D

MOUNT TAPE TO BE LABELED ON DEV=cuu.
cuu contains the channel, c, and the physical address, uu, of the unit that UTSSTI will use to start the I/O on the next tape. After this message appears, UTSSTI will wait for the operator to load a new tape before starting the I/O for the initialization.

UTSS0190 I

DEV=cuu, TAPE INITIALIZED WITH VOL=cccccc.
This informs the operator of the volume serial number (cccccc) of the tape which has just been successfully initialized on device cuu. This can serve as a guide, during a run to initialize a number of tapes, as to which exterior label goes on the newly initialized tape. No reply is required.

SYSOUT FILE ERROR MESSAGES

ERROR CODE **MESSAGE TEXT**

UTSS0000 TAPE INITIALIZE COMPLETED NORMALLY.
This message appears if and only if UTSSTI terminates without detecting any errors. It will be immediately preceded by message UTSS0110.

UTSS0014 AT LEAST ONE TAPE DRIVE //DEF CARD IS REQUIRED.
The user did not supply a correct DEFINE statement for the utility.

UTSS0024 AT LEAST ONE //PAR CARD IS REQUIRED.
The user omitted any //PAR cards.

UTSS0038 INDECIPHERABLE //PAR CARD ABORTS ANALYSIS.
An all blank or gibberish //PAR card was encountered.

UTSS0048 //PAR CARD CSD LENGTH BYTES MISSING.
System error in Common Stored Data Format Record Header.

UTSS0058 //PAR CARD DATA LENGTH=0.
System error in Common Stored Data Format Record Header.

UTSS0068 ID=OUTx DEVICE IS NOT TAPE.
The user specified an incorrect device type or address in his //DEF statement. (OUTx is filled in to correspond to the erroneous //DEFINE statement identifier.)

UTSS0078 ID=OUTx FAILED TO OPEN. RETURN INFORMATION=xxnn.
Open failed for reasons other than missing DEFINE statement. xxnn contains the return information from the OPEN service request. See the Data Management OPEN message where nn are the two rightmost hex status completion code digits.

UTSS0088 ID=OUTx FAILED TO CLOSE. RETURN INFORMATION=xxnn.
Close failure occurred for the tape unit specified on ID=OUTx. xxnn contains the return information from the CLOSE service request. See the Data Management CLOSE message where nn are the two rightmost hex status completion code digits.

UTSS0098 ID=OUTx INITIALIZE FAILED. RETURN INFORMATION=xxnn.
DEVICE=cuu. LABEL=cccccc.
An attempt to initialize a tape on the device cuu (channel and unit number) failed. The volume serial number which would have been written is supplied as cccccc. xxnn is the PCB return information field. See the I/O Control message where nn are the two rightmost hex status completion code digits.

UTSS0108 ACCEPT FAILURE. RETURN INFORMATION=xxnn.
The ACCEPT service request failed, yielding the return information provided in xxnn. See the I/O Control ACCEPT message where nn are the two rightmost hex status completion code digits.

ERROR CODE	MESSAGE TEXT
UTSS0112	LAST SUCCESSFULLY INITIALIZED TAPE HAD VOL=cccccc. If any tapes have been successfully initialized during the execution of UTSSTI, the volume serial number of the last one so initialized is inserted in this message. Z will contain the highest T returned so far. If Z is not zero, this is the last message. If Z is zero, message UTSS0000 follows it. This message does not appear if no tapes have been initialized.
UTSS0202	INITIALIZE NOT ATTEMPTED. The //PAR card above contained at least one error. Messages between the copy of the //PAR card and this message indicate those errors successfully detected.
UTSS0222	UNEXPECTED CONTINUATION. ACCEPT is not at end-of-file but the //PAR card above appeared to be the last in the string.
UTSS0232	EXPECTED CONTINUATION NOT RECEIVED. The last //PAR card terminated with a comma or a comma and a set of blanks.
UTSS0242	COMMA BEFORE 1ST KEYWORD. //PAR, keyword. . .is illegal.
UTSS0252	KEYWORD ILLEGAL. The only legal keyword for UTSSTI is LABEL.
UTSS0262	ILLEGAL CHAR(S) AFTER PARAMETER. Keyword=(subpars), is legal. Keyword=(subpars), key is illegal. Keyword-parameter, is legal. Keyword=parameter, key is not etc.
UTSS0272	AFTER PARAMETER. . . TOO MANY COMMAS. Keyword=parameter,, is illegal.
UTSS0282	INVALID PARAMETER. The volume identifier parameter was illegal.
UTSS0292	PARAMETER MISSING. The volume identifier parameter is missing.
UTSS0302	MISMATCHED PARENTHESIS. Self-explanatory.
UTSS0312	TOO MANY SUBPARAMETERS. At most, 3 subparameters are permitted.
UTSS0322	HIGHEST VOLUME SERIAL > 999999. For nnn not equal to 1, the highest volume serial number that would have been created exceeds 999999.

If any of the following error conditions occur, at most one of the following messages appears. This does not mean that the others are not applicable. In these, xxx contains '1st', '2nd', or '3rd'.

ERROR CODE	MESSAGE TEXT
UTSS0332	xxx SUBPARAMETER LENGTH ERROR. Legal subparameter lengths. 3rd $\leq$ 10 2nd $\leq$ 3 1st (if nnn $>$ 1) must = 6, otherwise 1 $\leq$ 1st $\leq$ 6
UTSS0342	xxx SUBPARAMETER TYPE INVALID. 2nd (if not omitted), must be numeric. 1st (if nnn $>$ 1) must be numeric.
UTSS0352	xxx SUBPARAMETER INVALID VALUE. 2nd (if not omitted) must be $\geq$ 1 and $\leq$ 999. 1st (if nnn $>$ 1) must be $\geq$ 000000 and $\leq$ 999999.

DISC INITIALIZE

The Disc Initialize utility program issues only console messages.

CONSOLE MESSAGES

The message text appears on the operator console. The error code is not displayed.

ERROR CODE	MESSAGE TEXT
UTSS1000	DISC INITIALIZE REV mm-dd-yy PARAMETERS. This is the header message that precedes the execution of the initialize program retrieval of console parameters. mm-dd-yy is the date of the last revision.
UTSS1052	INVALID REPLY. This message appears when an invalid parameter has been received by UTSSDI. The request for the parameter will be repeated.
UTSS1102	SYSTEM AREAS OVERLAP. PCAT=ccc,n CCAT=ccc,nn ALTADR=ccc. This message appears when the space for the PCAT, CCAT, or ALTTRKS overlap each other. Initialize program will request those parameters again.
UTSS1152	DEVICE LABEL EXISTS WITH VOLUME IDENT=x. . .x. CONTINLIB? This message occurs when the pack specified to be initialized already contains a valid device label. Reply Y to reinitialize the pack.
UTSS1252	DRIVE x CYL xxx TRK xx HARDWARE STATUS xxxx DURING I/O. This error occurs when an error status is returned to the UTSSDI during a disc I/O operation. This message invokes the RETRY? message; xxxx contains the IFA hardware status code (Table E-2).

ERROR CODE **MESSAGE TEXT**

UTSS1300 RETRY?
This message is used to indicate to the UTSSDI user that he should either retry an operation that had an error or terminate the program. Response is YES or NO.

UTSS1352 CARD READ FAILURE.
This message occurs when an error status is returned during a card input operation. This invokes the RETRY? message.

UTSS1402 INVALID ID.
This message appears when the first card or the first card after an EOF card is not a valid deck identifier. This message invokes the RETRY? message.

UTSS1452 INVALID DECK.
This message appears when any of the cylinder 0 update decks (AL, CS, or Diagnostics) contains more cards than allowed by the reserved area on disc. This message invokes the RETRY? message.

UTSS1508 EXCESSIVE DEFECTIVE TRACKS.
This message occurs when surface analysis detects more defective tracks than can be assigned to alternate tracks. This condition causes abnormal termination.

UTSS1550 DISC INITIALIZE COMPLETE ON VOLUME xxxxxx DRIVE x.
This is the UTSSDI completion message.

UTSS1600 DEFECTIVE TRACKS ASSIGNED TO ALTERNATE TRACKS.

```
CYL      HD          CYL      HD

XXX      XX          XXX      XX
 .        .           .        .
 .        .           .        .
 .        .           .        .
XXX      XX          XXX      XX
```

This table is printed after the UTSSDI completion message or when there are too many defective tracks.

REBUILD TRACK (UTSSRT) MESSAGES

Messages from this utility program appear in one of three places:

- Operator console
- SYSOUT file
- Listing of the entire track

CONSOLE MESSAGES

An invalid reply to these messages causes UTSSRT to repeat the message.

ERROR CODE	MESSAGE TEXT
UTSS2900	IS VOLUME cccccc A SYSTEM-RESIDENT PACK? REPLY Y OR N. UTSSRT is unable to determine from internal evidence whether volume cccccc is system-resident. Operator replies Y if the volume is system-resident and N if it is not.

If the user wants to assign an alternate track, message UTSS2910 appears; if he wishes to change records, message UTSS2920 appears. Either (or both) appears only after the track has been listed. The operator replies are CONT to continue execution and END to terminate execution.

If both messages appear, an operator reply of END to either message terminates execution without either user function being performed.

ERROR CODE	MESSAGE TEXT
UTSS2910	ALTERNATE TRACK ASSIGNMENT REQUESTED. REPLY 'CONT' OR 'END'. The user has specified ALTRAK=YES in his //PAR statements. UTSSRT is requesting permission to make the assignment. Reply: CONT or END.
UTSS2920	CORRECTIVE DATA TO BE APPLIED. REPLY 'CONT' OR 'END'. The user wants to change data on the track. UTSSRT is requesting permission to make the change. Reply: CONT or END.

SYSOUT FILE MESSAGES

Unless explicitly stated otherwise, messages from UTSSRT are displayed before the track is listed. Messages UTSS2074 and UTSS2502 through UTSS2742 appear on SYSOUT following the //PAR card in error. The final digit of the error code (0, 2, 4, 8) is explained below.

MESSAGE LEVEL	UTSSRT ACTION
0	Continue executing unless otherwise specified.
2	Continue //PAR statement analysis.
4	HALT
8	EHALT

ERROR CODE	MESSAGE TEXT
UTSS2000	NORMAL TERMINATION. This message appears after all requested user functions have completed normally and prior to final file closure.
UTSS2014	USER MUST SUPPLY DEFINE STATEMENT WITH ID=OUT. The required //DEFINE statement is missing.
UTSS2028	PAR CARD CSD FIELD HAS NO LENGTH BYTES. This message indicates an error in common stored data format of accepted //PAR card.
UTSS2038	PAR CARD DATA LENGTH=0. This message indicates an error in common stored data format of accepted //PAR card.
UTSS2044	EXECUTION NOT ATTEMPTED. This message occurs if //PAR statement errors whose highest T=2 are discovered.
UTSS2054	RE-RUN THIS DECK USING A DIFFERENT JOB NAME. During allocation of UTSSRT's scratch file, a duplicate file name was discovered in the central catalog but not in the pack catalog.
UTSS2064	ILLEGAL TO CHANGE RECORD SIZE UNLESS ALTRAK=YES. This message applies to end-of-file records. These are the *only* records whose data length can be changed by UTSSRT.
UTSS2074	//PAR CARD INCLUDED WITH NO IDENTIFIABLE KEYWORD= PARAMETER. ANALYSIS ABORTED. An indecipherable //PAR statement causes termination of //PAR statement scan.
UTSS2084	ILLEGAL TO ASSIGN AN ALTERNATE TO AN EXISTING ALTERNATE. The user has requested ALTRAK=YES but the track specified via TRACK= is an assigned alternate track.
UTSS2094	UTSSRT TERMINATED BY OPERATOR. This message occurs after the track printout and before any user-requested actions are taken. It is issued if the operator replies END to either of the console messages UTSS2910 or UTSS2920.
UTSS2108	IRRECOVERABLE DISC I/O ERROR RETURN INFORMATION = xxnn HARDWARE STATUS = xxxx, COMMAND FAILURE = op brs. This message can occur any time UTSSRT reads or writes from or to the disc. It indicates that an I/O operation terminated abnormally. xxnn contains the return information from the PCB. See the I/O Control message where nn are the two rightmost hex status completion code digits. xxxx contains the IFA hardware status code (see Table E-2), op indicates the I/O operation which failed and will be one of: SEEK, SEARCH, READ, READ W/O XFR, WRITE, or FORMAT WRITE. brs indicates the bit ring sync and will be blank or one of HOME ADR, RO COUNT, RN COUNT, KEY, DATA, CS DATA, or INV BRS (for an illegal bit ring sync).

ERROR CODE **MESSAGE TEXT**

UTSS2118

SCRATCH FILE WRITE FAILED. RETURN INFORMATION = xxnn.
An unsuccessful WRITE has been made to UTSSRT's scratch file occurred.
The nn contains the return information from the WRITE function.
See the I/O Control message where nn are the two rightmost hex status completion code digits.

UTSS2128

ABNORMAL RETURN FROM ALLOCATE RETURN INFORMATION = xxnn.
An unsuccessful attempt has been made to allocate UTSSRT's scratch file (for reasons other than those in messages UTSS2054 and UTSS2168). xxnn is the return information from the ALLOC function. See the Data Management ALLOC message where nn are the two rightmost hex status completion code digits.

UTSS2138

ABNORMAL RETURN FROM PURGE RETURN INFORMATION = xxnn.
An unsuccessful attempt has been made to purge UTSSRT's scratch file. This message occurs after any of the messages UTSS2000 through UTSS2128, UTSS2148 through UTSS2250. xxnn contains the return information from the PURGE function. See the Data Management PURGE message where nn are the two rightmost hex status completion code digits.

UTSS2148

ABNORMAL RETURN FROM CLOSE RETURN INFORMATION = xxnn.
An unsuccessful attempt has been made to close either ID=PRINTER or UTSSRT's scratch file. This occurs after any of messages UTSS2000 through UTSS2250. It contains the return information from the CLOSE function.
See the Data Management CLOSE message where nn are the two rightmost hex status completion code digits.

UTSS2158

SCRATCH FILE READ FAILED RETURN INFORMATION = xxnn.
An unsuccessful attempt has been made to read UTSSRT's scratch file. This occurs during record correction phase of UTSSRT. xxnn is the return information from READ. See the I/O Control message where nn are the two rightmost hex status completion code digits.

UTSS2168

SPACE IS INADEQUATE TO ALLOCATE SCRATCH FILE RETURN INFORMATION = xxnn.
The return information xxnn from the ALLOC function will indicate whether the space lack is on a pack, in a pack catalog, or in the central catalog. See the Data Management ALLOC message where nn are the two rightmost hex status completion code digits.

UTSS2174

PARTITION SPACE NOT LARGE ENOUGH TO PRINT ANY RECORDS ON TRACK.
The largest buffer available for the listing of the track is less than 2 bytes long.

UTSS2184

PARTITION SPACE NOT LARGE ENOUGH TO CORRECT RECORD(S).
The data length of at least one of the records to be corrected exceeds the buffer space available.

UTSS2194

PARTITION SPACE NOT LARGE ENOUGH TO COPY TRACK TO ALTERNATE.
For at least one record on the track, the available buffer space is less than the key length plus the data length of the record.

ERROR CODE **MESSAGE TEXT**

UTSS2204 CORRECTIVE DATA SUPPLIED FOR NON-EXISTENT RECORD xx.
The user has supplied (via a BADREC statement) a number, xx (in decimal), which does not match the record number in any count field on the track.

UTSS2214 CORRECTIONS PAST END OF RECORD-RECORD xx.
In a set of corrections for a given record xx (in decimal), the displacement within the record plus the number of bytes of corrective data exceeds the data length of the record.

UTSS2224 RECORD SIZE EXCEEDS TRACK CAPACITY -- RECORD xx.
For record xx (in decimal), two cases occur:

1. If the key length is nonzero, then the key length plus the data length exceeds 7249 bytes.
2. If the key length is zero, then the data length exceeds 7294 bytes.

This message indicates an erroneously written count field or, for MRX end-of-file records, an erroneously written end-of-file or an erroneous condition.

UTSS2234 NO ALTERNATE TRACKS ASSIGNABLE ON VOLUME cccccc.
During the scan to find an available alternate track, it was discovered that all possible alternates fall into one of three categories:

1. Already assigned.
2. Defective and thus unusable.
3. Home address, RO count, and/or data are unreadable.

UTSS2248 ON CYL ccc TRK xx HOME ADDRESS CONTAINS CYL bbb TRK yy.
This message indicates an erroneously written home address on the user-specified track. ccc or bbb are, in decimal, the cylinder numbers; xx and yy are, in decimal, the head numbers.

UTSS2250 CYLINDER ccc HEAD xx IS DEFECTIVE. UTSSRT WILL USE ALTERNATE CYLINDER bbb HEAD yy.
This message indicates that the listing of the track and corrections to records on the track will actually perform their I/O on the assigned alternate track. If ALTRAK=YES, however, this will be followed by message UTSS2084. ccc, xx, bbb, and yy are the same as specified in message UTSS2248.

The remaining messages (UTSS2502 through UTSS2742) relate to //PAR statement analysis.

ERROR CODE **MESSAGE TEXT**

UTSS2502 COMMA ILLEGAL BEFORE 1ST KEYWORD.
A comma appears after //PAR and before the first keyword on the card.

ERROR CODE	MESSAGE TEXT
UTSS2512	BADREC STATEMENT SEQUENCE ERROR. The BADREC statement in the //PAR statement contains a lower rrrr (record number) than the previous BADREC statement.
UTSS2522	KEYWORD SEQUENCE ERROR. A DATA parameter must follow a BADREC statement or another DATA statement. Otherwise too many DATA parameters follow a BADREC statement, or the nnn DATA parameters were interrupted by a different keyword.
UTSS2532	DATA KEYWORD NOT ALONE ON //PAR CARD. //PAR statements containing a DATA keyword must contain one and only one keyword.
UTSS2542	TOO MANY COMMAS AFTER PARAMETER. More than one comma separates a parameter and the keyword which follows it.
UTSS2552	COMMA MISSING AFTER PARAMETER. A comma must separate a parameter from the keyword which follows it.
UTSS2562	KEYWORD xxxxxx LENGTH ERROR. The only legal keywords for UTSSRT are from 4 to 6 bytes long.
UTSS2572	KEYWORD xxxxxx ILLEGAL. The keyword xxxxxx is not a legal keyword for UTSSRT.
UTSS2582	KEYWORD xxxxxx DUPLICATE. The keywords TRACK and ALTRAK must not be duplicated.
UTSS2592	KEYWORD TRACK IS MISSING. TRACK is the one required keyword.

In messages UTSS2612 through UTSS2672 below, xxxxxx contains the keyword whose parameter(s) is(are) in error. In messages UTSS2612 through UTSS2632, yyy contains 1st, 2nd, or 3rd in order to specify which subparameter is invalid.

ERROR CODE	MESSAGE TEXT
UTSS2612	xxxxxx yyy SUBPARAMETER MUST BE NUMERIC. The yyy subparameter contains a nonnumeric character.
UTSS2622	xxxxxx yyy SUBPARAMETER LENGTH ERROR. The yyy subparameter is either missing or too long.
UTSS2632	xxxxxx yyy SUBPARAMETER INVALID VALUE. The yyy subparameter value is too small or too large.
UTSS2642	xxxxxx PARAMETER MISMATCHED PAREN. A dangling left or right parenthesis follows keyword xxxxxx.

ERROR CODE	MESSAGE TEXT
UTSS2652	xxxxxx PARAMETER TOO cccc SUBPARAMETERS. cccc will contain MANY or FEW depending on whether there are too many or too few subparameters for keyword xxxxxx.
UTSS2662	xxxxxx PARAMETER IS MISSING. A blank or comma follows an equal sign, or a right parenthesis immediately follows a left parenthesis.
UTSS2672	xxxxxx PARAMETER LENGTH ERROR. The parameter has too many or too few characters.
UTSS2682	ALTRAK PARAMETER MUST BE YES OR NO.
UTSS2692	DATA PARAMETER MUST BE HEXADECIMAL.
UTSS2702	END-OF-CARD BEFORE END-OF-KEYWORD. A keyword did not terminate with an equal sign.
UTSS2712	END-OF-CARD BEFORE END-OF-PARAMETER. Card ended with an equal sign or a left parenthesis.
UTSS2722	ILLEGAL TO USE UTSSRT ON CYLINDER 0, TRACK 0. TRACK = (0,0) is illegal.
UTSS2732	ILLEGAL TO USE UTSSRT ON CYLINDER 0 OF A SYSRES PACK. TRACK = (0,n) ($1 \le n \le 19$) is illegal for a system-resident pack.
UTSS2742	EXPECTED CONTINUATION NOT RECEIVED. The last //PAR statement ended with a comma.

LISTING MESSAGES

In the printed output the count field appears before the listing of the data record. If the record is an end-of-file record, the actual data length is also shown. Certain remarks which may appear immediately after the line showing the count field are listed below.

MESSAGE TEXT

BURST CHECK OCCURRED WHILE READING DATA.
This record is either too long or too short.

END OF RECORD.
This message appears after the data has been listed.

$$\text{FAILURE ON}\left\{\begin{array}{l}\text{READ DATA}\\ \text{READ KEY}\\ \text{SEARCH COUNT}\end{array}\right\}\text{STATUS = xxxx.}$$

While attempting to read this record, an I/O error occurred. The IFA hardware status (see Table E-2) is given in xxxx.*

*No data is listed for the record in these cases.

	MESSAGE TEXT
	PARTITION SPACE IS NOT LARGE ENOUGH TO PRINT THIS RECORD. The record exceeds the buffer space available for the print routine.*
	THIS RECORD HAS NO DATA. This message occurs if the data length indicated in the count field is zero and there actually is no data.*

CHANGE VOLUME SERIAL NUMBER (UTSSCL) MESSAGES

Messages UTSS3018 through UTSS3118 appear on the SYSOUT file following the //PAR statement card containing the error(s). Error UTSS3068 applies to the previous //PAR statement if it did not terminate with a comma.

SYSOUT FILE MESSAGES

ERROR CODE	MESSAGE TEXT
UTSS3000	VOLUME SERIAL NUMBER xxxxxx HAS BEEN CHANGED TO yyyyyy. This message is output to both the console and the SYSOUT file upon normal termination of UTSSCL.
UTSS3018	PARAMETER CARD FORMAT ERROR – ONLY ONE COMMA MAY SEPARATE A PARAMETER FROM A KEYWORD. More than one comma separated a parameter from a keyword.
UTSS3028	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST BE DELIMITED BY AN EQUAL SIGN. While scanning for the end of a keyword, a space was found.
UTSS3038	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST PRECEDE AN EQUAL SIGN. An equal sign was found without a keyword preceding it.
UTSS3048	INVALID KEYWORD xxx. . . . The keyword xxx. . . . is invalid.
UTSS3058	DUPLICATE KEYWORD xxxxxxxx. Two keywords xxxxxxxx were entered via //PAR statements.
UTSS3068	A COMMA MUST SEPARATE PARAMETERS FROM KEYWORDS. A comma did not separate a parameter from a keyword.
UTSS3078	INVALID PARAMETER FOLLOWING KEYWORD xxxxxxxx. The parameter following the keyword xxxxxxxx contains an illegal character or its length is incorrect.

*No data is listed for the record in these cases.

ERROR CODE **MESSAGE TEXT**

UTSS3098

CONTINUATION WAS INDICATED BUT NO CONTINUATION CARD WAS FOUND.
The last //PAR statement ended with a comma.

UTSS3108

A LEFT PARENTHESIS WAS FOUND WITH A RIGHT PARENTHESIS.
This is an incorrect termination of subparameterization.

UTSS3118

REQUIRED KEYWORD xxxxxxxx WAS NOT FOUND.
A required keyword was not entered through a //PAR statement.

UTSS3138

A DISC DRIVE WAS NOT ASSIGNED TO THIS JOB.
The unit table does not contain a disc drive assigned to this job.

UTSS3148

IRRECOVERABLE I/O ERROR WHEN READING DEVICE LABEL. LAST COMMAND CODE = yy, RETURN INFORMATION = xxnn, RETURN HARDWARE STATUS = zzzz (see Table E-2).
An I/O operation failed to complete successfully when reading the device label. yy is a hexadecimal command code having the meaning listed below:

yy	Command Code
28	JUMP TO COMMAND
22	RESTORE
21	DCABLE
20	SEEK
08	SEARCH
02	READ
04	READ W/O XFER
01	WRITE
10	FORMAT WRITE

See the I/O Control message where nn are the two rightmost hex status completion code digits.

UTSS3158

IRRECOVERABLE I/O ERROR WHEN WRITING DEVICE LABEL. LAST COMMAND CODE = yy, RETURN INFORMATION = xxnn, RETURN HARDWARE STATUS = zzzz.
An I/O operation failed to complete successfully when writing the device label. Refer to Table E-2. In the message yy is a hexadecimal command code having the meanings listed below:

yy	Command Code
28	JUMP TO COMMAND
22	RESTORE
21	DCABLE
20	SEEK

ERROR CODE MESSAGE TEXT

yy	Command Code
08	SEARCH
02	READ
04	READ W/O XFER
01	WRITE
10	FORMAT WRITE

See the I/O Control messages where nn are the two rightmost hex status completion code digits.

UTSS3168

THE VOLUME xxxxxx MOUNTED VIA JCL, DOES NOT MATCH THE OLDVOL PARAMETER yyyyyy.
Either an incorrect volume was specified in the Control Language statement or an incorrect parameter followed OLDVOL. CATALOG DISPLAY (UTSSCD) MESSAGES.

CATALOG DISPLAY (UTSSCD) MESSAGES

The normal termination message is both printed on the output listing and sent to the console typewriter. Errors occurring when ID=PRINTER are sent to the console typewriter. All others are printed as part of the output listing. If the error code terminates in 8, the utility EHALTS. Codes less than 8 proceed to appropriate retry paths. Failure to write to the console typewriter causes an EHALT without a message.

CONSOLE AND OUTPUT LISTING MESSAGES

ERROR CODE MESSAGE TEXT

UTSS400X

END OF CATALOG DISPLAY.
This message is printed upon normal completion of UTSSCD. x will contain the highest return type discovered during normal processing (0, 2, or 4).

UTSS41x8

I/O ERROR: yyyyyy CODE=xxnn ID=ident.
Sent when an I/O operation fails to complete successfully. xxnn contains the return information field. ident contains the identifier of the file for which the error occurred.

x	yyyyyy
0	OPEN (Data Management)
1	READ (I/O Control)
2	WRITE (I/O Control)
3	CLOSE (Data Management)
4	CNTROL (for CONTROL macro) (Data Management)

See the I/O Control message or the Data Management message where nn are the two rightmost hex status completion code digits.

ERROR CODE **MESSAGE TEXT**

UTSS42x8

SR ERROR: yyyyyy CODE=xxnn.
This message is sent when a non-I/O service request completes abnormally. xxnn contains the return information field. For LOAD requests, see the LOADER message where nn are the two rightmost hex status code digits. nn are zeros for the other requests.

x	yyyyyy
0	SDATE
1	TIME
2	LOAD
3	GETADR

UTSS4508

BB=0 REQUESTED.
This message is sent by the root module GET subroutine if entry is made requesting a block number of zero. It indicates an error in the catalog structure.

UTSS4518

RECORD END PAST END-OF-BLOCK.
This message is sent by the root module GET subroutine. It indicates an error in CSDF structure.

UTSS4528

REC HEADER PAST END-OF-BLOCK.
This message is sent by the root module GET subroutine. It indicates an error in CSDF structure.

UTSS4538

POINTERS PAST END-OF-RECORD.
This message is sent by the pack catalog overlay or the central catalog overlay. It indicates an error in the name element of the catalog concerned.

UTSS4548

SPACE ELEMENT POINTER ABSENT.
This message is sent by the space map overlay. It indicates an error in the name element entry of the space map in the pack catalog.

UTSS4558

TOTAL SEGMENT COUNT < 0.
This message is sent by the space map overlay. It indicates an error in the space element entry of the space map in the pack catalog.

UTSS456x

NO SEGMENT DESCRIPTIONS.
This message is sent by the pack catalog overlay or the space map overlay. It indicates an error in the space element record.
x = 4 for the first such error for the same IDENT= and overlay. If another such error occurs for the same IDENT= and overlay, x = 8.

UTSS4578

MISSING SPACE OR CONT ELEM.
This message is sent by the pack catalog or the space map overlay. It indicates an error in the space element record.

UTSS458x

NO VOLUME DESCRIPTIONS.
This message is sent by the central catalog overlay. It indicates an error in the volume element record.

x = 4 for the first such error

x = 8 for all other errors

ERROR CODE	MESSAGE TEXT
UTSS4598	MISSING VOLUME OR CONT ELEM. This message is sent by the central catalog overlay. It indicates an error in the volume element record.
UTSS4608	BYTE COUNT INPUT ERROR. This message is sent by the root module READ subroutine if the residual byte count is not zero.
UTSS4618	FDT NOT FOUND AFTER OPEN. Sent by root module. Self-explanatory.
UTSS4628	NO //DEF STATEMENT FOR PCAT1 OR CCAT. No //DEFINE statements with ID=PCAT1 or ID=CCAT appeared in the JCL. At least one such statement is required.
UTSS4638	I/O ERROR: HASH TABLE CODE=xxnn ID=PCATy. When trying to read the hash table in the file whose ID=PCATy, an I/O error occurred. xxnn is the return information field from the READ service request. y will be a number from 1 to 7. See the I/O Control message, where nn are the two rightmost hex status completion digits.
UTSS4648	I/O ERROR: HASH TABLE CODE=xxnn ID=CCAT. When trying to read a hash table block from the file whose ID=CCAT, an I/O error occurred. xxnn is the return information field from the READ service request. See the I/O Control message where nn are the two rightmost hex status completion digits.

PURGE FILE (UTSSPF) MESSAGES

All messages from the purge file are routed to the user via SYSOUT. Prior to each message (except for UTSS5108, UTSS5208, and UTSS5218) appears a copy of the //PAR statement to which it pertains.

Fatal errors have the reference number UTSS5xx8. These errors cause an EHALT. Other errors proceed to ACCEPT and scan the next //PAR statement. No further analysis of a //PAR statement takes place once an error is discovered.

Unsuccessful attempts to write to SYSOUT cause an EHALT with no error message.

SYSOUT FILE MESSAGES

ERROR CODE	MESSAGE TEXT
UTSS5000	PURGE SUCCESSFUL. The file shown in the preceding //PAR statement has been successfully purged from the system.

ERROR CODE **MESSAGE TEXT**

UTSS5108 SR ERROR: ACCEPT RETURN INFORMATION = xxnn.
The ACCEPT macro returned the abnormal completion code shown in xxnn. See the Control Program ACCEPT message where nn are the two rightmost hex status completion code digits.

UTSS511x SR ERROR: PURGE RETURN INFORMATION = xxnn.
The attempt to purge the file shown in the preceding //PAR statement resulted in the abnormal completion code shown in xxnn.

x = 8 if irrecoverable I/O error

x = 4 otherwise

See the Data Management PURGE message where nn are the two rightmost hex status completion code digits.

UTSS5208 //PAR CARD CSD FIELD HAS NO LENGTH BYTES.
This message indicates an error in common stored data format of the //PAR statement.

UTSS5218 //PAR CARD DATA LENGTH = 0.
This message indicates an error in common stored data format of the //PAR statement.

UTSS5228 NO IDENTIFIABLE KEYWORD = PARAMETER. ANALYSIS ABORTED.
An indecipherable //PAR statement caused termination of //PAR statement scan.

The following messages all indicate that the user has made an error in formating his //PAR statements. These messages also contain the words PURGE NOT ATTEMPTED between the error code field and the remaining text.

ERROR CODE **MESSAGE TEXT**

UTSS5302 COMMA BEFORE 1ST KEYWORD – ILLEGAL.
There must be no commas between //PAR and PURGE.

UTSS5312 FOLLOWING PARAMETER(S). . .TOO MANY COMMAS.

Legal Forms:

```
PURGE=filename
PURGE=filename,
PURGE=(parameters)
PURGE=(parameters),
```

Illegal Forms:

```
PURGE=filename,,
PURGE=(parameters),,
```

ERROR CODE **MESSAGE TEXT**

UTSS5322 FOLLOWING PARAMETER(S). . .ILLEGAL CHARACTER.

Illegal Forms:

PURGE=filename,P

UTSS5332 KEYWORD LENGTH ERROR.
The only valid keyword PURGE is 5 characters long.

UTSS5342 xxxxxx ILLEGAL; ONLY 'PURGE' IS ALLOWED.
xxxxxx is the illegal keyword.

UTSS5352 MISMATCHED PARENTHESIS.
Both halves of a pair of parentheses must appear on the same card.

UTSS5362 TOO MANY SUBPARAMETERS.
A maximum of 4 subparameters is allowed.

UTSS5372 END-OF-CARD BEFORE END-OF-KEYWORD.
End-of-card was encountered before the equal sign.

UTSS5382 END-OF-CARD BEFORE END-OF-PARAMETER.
The equal sign or a left parenthesis was the last character on the card.

UTSS5392 PARAMETER LENGTH ERROR.

Subparameter	Maximum Byte Length
filename	17
msc	4
NO	2
P	1

UTSS5402 1ST SUBPARAMETER MUST BE PRESENT.
The filename is required.

UTSS5412 1ST SUBPARAMETER CONTAINS ILLEGAL CHARACTER.
Legal characters for filename include A-Z, 0-9, the dollar sign, and the dash.

UTSS5422 3RD SUBPARAMETER MUST BE OMITTED OR 'NO'.

UTSS5432 4TH SUBPARAMETER MUST BE OMITTED OR 'P'.

UTSS5442 }
UTSS5452 } END-OF-JOB WILL PURGE { $------SYSIN / $------SYSOUT }.
An attempt to purge the temporary file $------SYSIN, jobname or $------SYSOUT, jobname before end of job would cause a disastrous error.

ALLOCATE FILE (UTSSAF) MESSAGES

All messages are routed to the user via SYSOUT. Included in the SYSOUT string is a copy of each //PAR statement submitted. Messages where EEE=630 and above concern the contents of the //PAR statements. These messages follow the copy of the //PAR statements containing the error(s). Messages where EEE=600, 601, or 611 follow the complete set of //PAR statement copies. Messages where EEE=610, 620, 621 appear without a copy of the //PAR statement.

Fatal errors have T=8. These errors cause UTSSAF to EHALT. T=4 errors cause UTSSAF to HALT. T=2 errors (except EEE=601) proceed to further analysis of the current //PAR statement or to the ACCEPT macro and scan the next //PAR statement. Unsuccessful attempts to write to SYSOUT cause an EHALT with no error message.

SYSOUT FILE MESSAGES

ERROR CODE	MESSAGE TEXT
UTSS6000	ALLOCATION SUCCESSFUL. The file pair described in the //PAR statement string has been allocated successfully.
UTSS6012	ALLOCATION NOT ATTEMPTED. Because of errors in //PAR statements, no attempt was made to allocate the file pair.
UTSS6108	SR ERROR: ACCEPT RETURN INFORMATION = xxnn. See the Control Program ACCEPT message where nn are the two rightmost hex status completion code digits.
UTSS611x	SR ERROR: ALLOC RETURN INFORMATION = xxnn. The attempt to allocate the file pair was unsuccessful. x = 8 if irrecoverable I/O error x = 4 for all other conditions See the Data Management ALLOC message text where nn are the rightmost two hex status completion code digits.
UTSS6208	//PAR CARD CSD FIELD HAS NO LENGTH BYTES. This message indicates an error in common stored data format of the //PAR statement.
UTSS6218	//PAR CARD DATA LENGTH = 0. This message indicates an error in common stored data format of the //PAR statement.
UTSS6308	COMMA BEFORE 1ST KEYWORD – ILLEGAL. There must be no commas between //PAR and the first keyword on the //PAR statement.

ERROR CODE	MESSAGE TEXT
UTSS6312	TOO MANY COMMAS BETWEEN PARAMETER AND KEYWORD. Legal Form: keyword=par, keyword=par Illegal Form: keyword=par,,keyword. . .
UTSS6322	COMMA MISSING BETWEEN PARAMETER AND KEYWORD. Illegal Form: Keyword=par keyword. . .

In the following messages the field xxxxxxxx contains the keyword (a) which is in error or (b) whose parameter is in error.

ERROR CODE	MESSAGE TEXT
UTSS6332	KEYWORD xxxxxxxx LENGTH ERROR. If xxxxxxxx is blank, then a keyword of length 0 was found. If xxxxxxxx contains characters, keyword length exceeded 8. The first 8 bytes are reproduced.
UTSS6342	KEYWORD xxxxxxxx ILLEGAL. The keyword shown in xxxxxxxx is not a valid keyword for UTSSAF.
UTSS6352	KEYWORD xxxxxxxx DUPLICATE. Duplication of keywords in the same execution is not permitted by UTSSAF.
UTSS6362	xxxxxxxx PARAMETER: PARENTHESIS ILLEGAL. UTSSAF does not allow parentheses around its parameters.
UTSS6372	xxxxxxxx PARAMETER: LENGTH ERROR. The parameter associated with the keyword xxxxxxxx has either too many or too few characters.
UTSS6382	xxxxxxxx PARAMETER: MUST BE ALPHABETIC.
UTSS6392	xxxxxxxx PARAMETER: MUST BE 'YES' OR 'NO'.
UTSS6402	xxxxxxxx PARAMETER: MUST BE NUMERIC.
UTSS6412	xxxxxxxx PARAMETER: VALUE OUT OF RANGE. The value of a numeric parameter is either too large or too small to be acceptable.
UTSS6422	$\left\{\begin{array}{l}\text{FILENAME}\\ \text{FIL2}\end{array}\right\}$ PARAMETER: CONTAINS ILLEGAL CHARACTER(S). Legal characters for filenames include A-Z, 0-9, the dollar sign, and the dash.

ERROR CODE	MESSAGE TEXT
UTSS6432	ORG PARAMETER: MUST BE 'S' OR 'R'.
UTSS6442	REQUIRED PARAMETER(S) MISSING: list. At least one of the required parameters is missing. The contents of list are the keywords expected but not found.
UTSS6452	WHEN CAT=NO, ID=YES IS REQUIRED.
UTSS6462	END-OF-CARD BEFORE END-OF-KEYWORD. End-of-card was encountered before the equal sign.
UTSS6482	EXPECTED CONTINUATION NOT RECEIVED. The last //PAR statement ended with a comma.
UTSS6498	NO IDENTIFIABLE KEYWORD = PARAMETER. ANALYSIS ABORTED. An indecipherable //PAR statement causes termination of //PAR statement scan.

STAND-ALONE DISC-TO-DISC COPY MESSAGES

The normal termination message and the set of parameter request messages are included in the utility description. Below are found the error messages, classified according to the programming phase where they occur.

CONSOLE MESSAGES

The rules listed below pertain to the console messages.

1. In all messages where the words DRIVE x appear, x contains the disc drive number.

2. In all messages where the words STATUS xxxx appear, xxxx contains the IFA Hardware Status (see Table E-2).

3. In the messages containing the words ATTN=yyyy, yyyy contains the IFA attention and PDA status (see Table E-1).

4. When CYL ccc TRK hh appears in a message, ccc and hh are, respectively, the cylinder and head numbers (in decimal) of the track where an error has occurred. If REC nn also appears, nn is (in decimal) the record number on that track.

5. The message text appears on the operator console. The error code is not displayed.

MESSAGES OCCURRING DURING PARAMETER INPUT AND VALIDATION

ERROR CODE **MESSAGE TEXT**

UTSS7012

INVALID REPLY.
This message occurs when parameter format or content is invalid. It is followed by a repeat of the request for the parameter.

UTSS7022

REPLY CANCELLED.
This message results from the CAN, ENQ, or NAK termination characters. It is followed by a repeat of the request for the parameter.

MESSAGES OCCURRING DURING VALIDATION OF DISC DRIVE NUMBERS AND STATUS

ERROR CODE **MESSAGE TEXT**

UTSS7102

WAITING FOR DRIVE x TO COME ONLINE.
This is an informative message. The operator should make sure drive is started, enabled, and correctly mounted. Once the drive is up, either the next parameter request will appear or one of the other error messages.

$\left.\begin{array}{l}\text{UTSS7112}\\ \text{UTSS7122}\end{array}\right\}$

DRIVE x NOT MOUNTED FOR $\left\{\begin{array}{l}\text{READ-ONLY}\\ \text{READ/WRITE}\end{array}\right\}$.
RETRY;
After correcting the error, the operator should respond to the message. A response of Y or YES reselects the same drive and puts it through the tests. A reply of N or NO (the default case) causes the program to reissue the FROM DRIVE= or TO DRIVE= message.

UTSS7132

OUTPUT DRIVE=INPUT DRIVE.
MORE?
This message occurs if, after issuing TO DRIVE=, the operator has given the same drive number for both the input and output packs. A reply of Y or YES to MORE? will cause the FROM DRIVE= message to reappear. The default (or N or NO) will terminate the program normally.

UTSS7204

DRIVE x HARDWARE STATUS xxxx DURING SELECT.
RETRY?
The drive select procedure has detected an error in the IFA returned hardware status. A Y or YES response causes UTSSDD to reselect the same drive and put it through the status testing procedure. A N or NO (the default) reply causes UTSSDD to resend the appropriate FROM DRIVE= or TO DRIVE= message. See also the same message in the Messages Occurring During the Disc Copy Phase section of UTSSDD.

MESSAGES OCCURRING DURING VALIDATION OF VOLUME IDENTIFIERS

In addition to the special messages below, any of the standard I/O error messages can also occur.

ERROR CODE	MESSAGE TEXT
UTSS7144	DRIVE x VOLUME ID SPECIFIED NOT = ACTUAL VOLUME ID xxxxxx. The volume identifier input through the console does not match the one in the device label on drive x. xxxxxx contains the identifier from the device label. If all characters in the device label volume identifier are legal, UTSSDD resends the appropriate FROM DRIVE= or TO DRIVE= message. If any character(s) in the device label are illegal, however, a ? will be present in the field xxxxxx and the following message will be sent.
UTSS7152	? MEANS ILLEGAL CHARACTER(S) IN DEVICE LABEL VOLUME ID. CONTINUE? A Y or YES reply will proceed as if the identifier were legal; i.e., the next normal parameter request message is sent. A N or NO (the default) results in the MORE? message.
UTSS7408	DRIVE x CYL 0 TRK 0 – CANNOT READ DEVICE LABEL-STATUS xxxx-FATAL ERROR. The program halts. One means of recovery is to reinitialize cyl 0, track 0.

MESSAGE THAT MIGHT OCCUR DURING VALIDATION OF LIMITS

ERROR CODE	MESSAGE TEXT
UTSS7162	UPPER LIMIT < LOWER. This error causes UTSSDD to return to reissue the LOWER LIMITS= message.

MESSAGES OCCURRING DURING THE DISC COPY PHASE OF UTSSDD

These errors can occur on any track.

ERROR CODE	MESSAGE TEXT
UTSS7204	DRIVE x HARDWARE STATUS xxxx DURING SELECT. RETRY? Although identical in form to the message occurring during validation of disc drive numbers and status, the replies to this message have the following effect. A Y or YES initiates the retrial of the select-drive process. A N or NO (default) causes abnormal termination.

ERROR CODE	MESSAGE TEXT
UTSS7214	DRIVE x CYL ccc TRK hh HARDWARE STATUS xxxx DURING SEEK. RETRY? A Y or YES response causes UTSSDD to retry the seek; a N or NO (default) causes abnormal termination.
UTSS7224	DRIVE x CYL ccc TRK hh REC nn HARDWARE STATUS xxxx DURING I/O. RETRY? A Y or YES reply will generate a retry of the unsuccessful I/O operation. An N or NO response (the default) aborts the program. (See also Other Special-Purpose I/O Error Messages.)
UTSS7414 UTSS7424	DRIVE x CYL ccc TRK hh CANNOT READ $\left\{\begin{matrix}\text{HOME ADDRESS}\\ \text{RECD 0 COUNT}\end{matrix}\right\}$. STATUS xxxx. RETRY? If the operator replies Y or YES, UTSSDD retries the read. If the operator replies N or NO, the program terminates abnormally.
UTSS7438	DRIVE x CYL ccc TRK hh ATTN=yyyy BUT STATUS xxxx DURING SEEK-FATAL ERROR. The action recommended when this error occurs is to request a dump and call your service representative.
UTSS7448	DRIVE x CYL ccc TRK hh REC nn READ-NO SYNC COMPARE-FATAL ERROR. The count field previously read in indicated that (key and/or) data should follow, but none is present. The action recommended when this error occurs is to rebuild the track.

MESSAGES CONCERNING DEFECTIVE TRACKS

ERROR CODE	MESSAGE TEXT
UTSS7234	DRIVE x CYL ccc TRK hh DEFECTIVE – WILL USE ALT CYL ccc TRK hh. CONTINUE? A response of Y or YES will cause UTSSDD to go ahead and use the assigned alternate track (either input or output). A response of N or NO (the default) will send the end-of-run MORE? message.
UTSS7242	DRIVE x CYL ccc TRK hh DEFECTIVE BUT NO ALTERNATE ASSIGNED TRACK WILL BE BYPASSED. This message simply informs the operator of the automatic action that UTSSDD will take and where the defective track exists. After the copy is complete, the user would be well advised to reinitialize the defective track.
UTSS7458	DRIVE x CYL ccc TRK hh DEFECTIVE-FATAL ERROR. Defective tracks cause abnormal termination for any disc pack if either an alternate track assigned to a defective primary is itself a defective track or the defective track is cylinder 0, head 0. In addition, for Memorex system-resident packs, defective control storage data tracks are also cause for abnormal termination.

OTHER SPECIAL-PURPOSE I/O ERROR MESSAGES

ERROR CODE	MESSAGE TEXT
UTSS7224	DRIVE x CYL 0 TRK 0 REC 3 HARDWARE STATUS xxxx DURING I/O. RETRY? The replies to this message will do the same as in the more general case. However, if this is the output drive of a Memorex pack, this could mean that record 4 (the alternate track assignment table) is missing. This is the only case where the record number does not reflect reality.
UTSS7254	DRIVE x CYL 0 TRK hh – CANNOT READ C.S. DATA W/O BURST CK. CONTINUE? This message indicates that the length of the control storage data on the input pack is not a multiple of 512. The reply Y or YES will cause a full 6656 bytes to be written to the track of the output pack. A N or NO response (the default) proceeds to the end-of-run query MORE?
UTSS7468	DRIVE x CYL 0 TRK 0 – NO Rn COUNT – FATAL ERROR. This message applies only to a Memorex pack, track 0. If n=0, 1, or 2, it indicates that the input pack was not properly initialized or that track 0 has been at least partially destroyed.

CATALOG AND UNCATALOG (UTSSCU) MESSAGES

Messages from this utility program appear on the operator console and on the SYSOUT file.

CONSOLE AND SYSOUT FILE MESSAGES

ERROR CODE	MESSAGE TEXT
UTSS8000	MOUNT VOLUME vvvvvv ON xxx. The operator must mount volume vvvvvv on drive xx.
UTSS8000	ALL REQUESTED VOLUMES MOUNTED? Reply Y or YES if the volume has been mounted. Reply N or NO if the volume has not been mounted. Any other answer results in a repeat of the message.
UTSS8000	WRONG VOLUME MOUNTED. The operator must mount the correct volume.
UTSS8208	ABNORMAL RETURN FROM SYSOUT WRITE. RETURN INFORMATION = xxnn. The program was unable to write to SYSOUT. See the I/O Control message where nn are the two rightmost hex status completion code digits.

SYSOUT FILE MESSAGES

ERROR CODE **MESSAGE TEXT**

UTSS800X

NORMAL TERMINATION.
UTSSCU has completed normally. X is the highest message type code encountered during normal processing.

UTSS8010

xxxx UNCATALOGED OR PAIRED FILES xxxxx AND yyyyy UNCATALOGED.

UTSS8020

UNABLE TO OPEN CENTRAL CATALOG. RETURN INFORMATION = xxnn.
See the Data Management OPEN message where nn are the two rightmost hex status completion code digits.

UTSS8032

PAIRED FILES xxxxx AND ???? NOT UNCATALOGED. PAIRED FILE POINTER NOT FOUND IN CENTRAL CATALOG NAME ENTRY.

UTSS8042

xxxxx NOT UNCATALOGED. UNABLE TO OPEN. RETURN INFORMATION = xxnn.
PAIRED FILES xxxxx AND yyyyy NOT UNCATALOGED.
UNABLE TO OPEN zzzzz. RETURN INFORMATION = xxnn.
The program was unable to open the file (xxxxx) to be uncataloged. For paired files, file zzzzz is the file that could not be opened. See the Data Management OPEN message where nn are the two rightmost hex status completion code digits.

UTSS8052

xxxxx NOT UNCATALOGED. CONTINUATION ELEMENT POINTER IN CENTRAL CATALOG BLOCK 1 DESTROYED.
PAIRED FILES xxxxx AND yyyyy NOT UNCATALOGED. CONTINUATION ELEMENT POINTER IN CENTRAL CATALOG BLOCK 1 DESTROYED.
The pointer ID for continuation blocks was not found in block 1 of the central catalog.

UTSS8062

xxxxx NOT UNCATALOGED. FILE NAME zzzzz NOT FOUND IN CENTRAL CATALOG.
PAIRED FILES xxxxx AND yyyyy NOT UNCATALOGED. FILE NAME zzzzz NOT FOUND IN CENTRAL CATALOG.
File name zzzzz was not found in the central catalog.

UTSS8072

xxxxx NOT UNCATALOGED. UNABLE TO FIND INDEX FILE'S POINTER TO INFORMATION FILE.
The pointer ID for the information file was not found in the central catalog name element of the file.

UTSS8082

UNABLE TO UNCATALOG FILES SPECIFIED. CENTRAL CATALOG VOLUME MUST BE THE SYSTEM RESIDENT VOLUME.
The central catalog volume is not the system resident volume. None of the specified files are uncataloged.

ERROR CODE	MESSAGE TEXT
UTSS8098	BLOCK I/O ERROR READING CENTRAL CATALOG BLOCK. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits. If the message indicates invalid block number, the central catalog thread may be broken; otherwise an invalid pointer (BBR) was encountered.
UTSS8108	BLOCK I/O ERROR WRITING CENTRAL CATALOG BLOCK. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS8118	BLOCK I/O ERROR READING PACK CATALOG ON BLOCK ON VOLUME vvvvv. RETURN INFORMATION = xxnn. See the message in the I/O Control section where nn are the two rightmost hex status completion code digits. If an invalid block number is indicated by return information, the pack catalog thread may be broken or an invalid pointer (BBR) may have been encountered.
UTSS8128	BLOCK I/O ERROR WRITING PACK CATALOG BLOCK ON VOLUME vvvvv. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS8138	BLOCK I/O ERROR READING SYSIN BLOCK. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS8142	xxxxx NOT UNCATALOGED. VOLUME vvvvv NOT MOUNTED. The operator answered NO to a volume mount request.
UTSS8158	PHYSICAL I/O ERROR ATTEMPTING TO READ VOLUME LABEL ON VOLUME vvvvv. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS8168	PHYSICAL I/O ERROR ATTEMPTING TO READ PACK CATALOG FDT ON VOLUME vvvvv. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS8172	xxxxx NOT UNCATALOGED. FILE NAME NOT FOUND IN PACK CATALOG ON VOLUME vvvvv. PAIRED FILES xxxxx AND yyyyy NOT UNCATALOGED. FILE NAME zzzzz NOT FOUND IN PACK CATALOG ON VOLUME vvvvv. The file name xxxxx (if paired files, zzzzz) was not found in the pack catalog on volume vvvvv.

ERROR CODE	MESSAGE TEXT
UTSS8182	xxxxx NOT UNCATALOGED. //DEF STATEMENT VOLUME ID(S) DO NOT MATCH CENTRAL CATALOG VOLUME ID(S). PAIRED FILES xxxxx AND yyyyy NOT UNCATALOGED. //DEF STATEMENT VOLUME ID(S) DO NOT MATCH CENTRAL CATALOG VOLUME ID(S). If volumes were specified on the //DEF statement, they do not match the volumes in the central catalog entry of the file.
UTSS8190	xxxxx CATALOGED. PAIRED FILES xxxxx AND yyyyy CATALOGED. The file(s) have been cataloged.
UTSS8212	xxxxx NOT CATALOGED. DUPLICATE NAME IN CENTRAL CATALOG. PAIRED FILES xxxxx AND yyyyy NOT CATALOGED. DUPLICATE NAME zzzzz IN CENTRAL CATALOG. A duplicate name xxxxx orzzzzz was found in the central catalog.

UTSS8222	xxxxx NOT CATALOGED. STATUS=OUTPUT WAS NOT SPECIFIED ON DEFINE STATEMENT.
	PAIRED FILES xxxxx AND yyyyy NOT CATALOGED. STATUS=OUTPUT WAS NOT SPECIFIED ON //DEFINE STATEMENT.
	STA = (P,O) was not specified on the //DEF statement.

UTSS8232	xxxxx NOT CATALOGED. CONTINUATION ELEMENT POINTER IN CENTRAL CATALOG BLOCK 1 DESTROYED. PAIRED FILES xxxxx AND yyyyy NOT CATALOGED. CONTINUATION ELEMENT POINTER IN CENTRAL CATALOG BLOCK 1 DESTROYED. The pointer ID for continuation blocks was not found in block 1 of the central catalog.
UTSS8242	xxxxx NOT CATALOGED. FILE NAME NOT FOUND IN PACK CATALOG ON VOLUME vvvvv. PAIRED FILES xxxxx AND yyyyy NOT CATALOGED. FILE NAME zzzzz NOT FOUND IN PACK CATALOG ON VOLUME vvvvv. The file name xxxxx (if paired files, zzzzz) was not found in the pack catalog on volume vvvvv.
UTSS8252	xxxxx NOT CATALOGED. VOLUME vvvvv NOT MOUNTED. PAIRED FILES xxxxx AND yyyyy NOT CATALOGED. VOLUME vvvvv NOT MOUNTED. The operator answered NO to a volume mount request.
UTSS8262	PAIRED FILES xxxxx AND yyyyy NOT CATALOGED. PAIRED FILE POINTER NOT FOUND IN PACK CATALOG NAME ENTRY ON VOLUME vvvvvv. The pointer ID to a paired file was not found in the pack catalog name element of the file on volume vvvvvv.
UTSS8278	PHYSICAL I/O ERROR ATTEMPTING TO READ VOLUME LABEL ON VOLUME vvvvvv. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.

ERROR CODE	MESSAGE TEXT
UTSS8282	xxxxx NOT CATALOGED. UNABLE TO OBTAIN SUFFICIENT CENTRAL CATALOG BLOCKS. PAIRED FILES xxxxx AND yyyyy NOT CATALOGED. UNABLE TO OBTAIN SUFFICIENT CENTRAL CATALOG BLOCKS.
UTSS8308	PHYSICAL I/O ERROR ATTEMPTING TO READ PACK CATALOG FDT ON VOLUME vvvvvv.
UTSS8318	BLOCK I/O ERROR READING CENTRAL CATALOG BLOCK. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits. If the I/O message indicates invalid block number, the central catalog thread may be broken or an invalid pointer (BBR) was encountered.
UTSS8328	BLOCK I/O ERROR WRITING CENTRAL BLOCK RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS8338	BLOCK I/O ERROR READING PACK CATALOG BLOCK ON VOLUME vvvvvv. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits. If the I/O message indicates invalid block number, the pack catalog thread may be broken or an invalid pointer (BBR) was encountered.
UTSS8348	BLOCK I/O ERROR WRITING PACK CATALOG BLOCK ON VOLUME vvvvvv. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS8358	BLOCK I/O ERROR READING SYSIN BLOCK. RETURN INFORMATION = xxnn. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS8368	xxxxx NOT CATALOGED. NO VOLUMES WERE SPECIFIED ON //DEF STATEMENTS. PAIRED FILES xxxxx AND yyyyy NOT CATALOGED. NO VOLUMES WERE SPECIFIED ON //DEF STATEMENTS.
UTSS8372	NO UNCATALOGING DONE IN THIS EXEC. No //DEFINE statements with ID=UNCTLGxx were specified during this execution. No uncataloging was attempted.
UTSS8382	NO CATALOGING DONE IN THIS EXEC. No //DEFINE statements with ID=CTLGxx were specified during this execution. No cataloging was attempted.

LOAD UNIVERSAL CHARACTER SET

LOADUCS SERVICE REQUEST RETURN CODES

Hex Status Code	Meaning
2E01	Device address was not found in unit table.
2E02	Device type is not a printed supported by this utility.
2E03	No buffer address has been specified and the image name is not recognized.
2E04	Blank or Null found in image.
2E05	The member was not found in $NUCLIB.
2E06	Abnormal return from PIO on UCS Load command.
2E07	The UCS feature is not installed.

CONSOLE MESSAGES

UTSSLU sends the following messages to verify that the print chain/train is mounted prior to UCS Load, and that a visual scan of the printed buffer was satisfactory:

ERROR CODE	MESSAGE TEXT
UTSS9260	MOUNT PRINTER ARRANGEMENT XXXX ON PRINTER NNN. IS ARRANGEMENT MOUNTED? When VERification is requested, the utility program asks the above question to ensure that the proper print chain or train is installed on the 1403 prior to issuing the LOADUCS service request. The options are simply 'YES' or 'NO'. If 'YES' is specified, execution proceeds normally. 'NO' forces step termination.
UTSS9270	VERIFY UCS IMAGE XXXX ON DEVICE NNN. IS IMAGE CORRECT? After the UCS load has been performed and the buffer has been printed, if VERification is requested, the above message is displayed to the console requesting visual verification by the operator. The reply may be 'YES' or 'NO'. A reply of 'YES' will cause the utility to normally terminate. 'NO' will allow the operator to mount another cartridge and retry after the following message has been displayed.
UTSS9280	IS RETRY DESIRED? A reply of 'NO' will cause step abort. 'YES' sets the utility back to the point of which it requests the cartridge to be mounted.

ERROR CODE	MESSAGE TEXT
UTSS9292	INVALID REPLY. REPLY MUST BE YES, Y, NO, N. This message is displayed in response to an incorrect reply to one of the above messages.

SYSOUT FILE MESSAGES

ERROR CODE	MESSAGE TEXT
UTSS9018	PARAMETER CARD FORMAT ERROR – ONLY ONE COMMA MAY SEPARATE A PARAMETER FROM A KEYWORD.
UTSS9028	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST BE DELIMITED BY AN EQUAL SIGN.
UTSS9038	PARAMETER CARD FORMAT ERROR – A KEYWORD MUST PRECEDE AN EQUAL SIGN.
UTSS9048	INVALID KEYWORD.
UTSS9058	DUPLICATE KEYWORD. Only one UCS keyword is allowed.
UTSS9068	A COMMA MUST SEPARATE PARAMETERS FROM KEYWORDS.
UTSS9078	INVALID PARAMETER FOLLOWING KEYWORD.
UTSS9088	TOO MANY SUBPARAMETERS ENTERED FOR KEYWORD.
UTSS9098	CONTINUATION WAS INDICATED BUT NO CONTINUATION CARD WAS FOUND.
UTSS9108	A LEFT PARENTHESIS WAS FOUND WITHOUT A RIGHT PARENTHESIS.
UTSS9118	REQUIRED KEYWORD UCS WAS NOT FOUND.
UTSS9128	ABNORMAL RETURN FROM ACCEPT MACRO.
UTSS9138	CHARACTER SET IMAGE NAME MUST BE SUPPLIED. This error message is printed if the first positional subparameter is not supplied.
UTSS9148	INVALID $\left\{\begin{matrix}\text{FOLDING}\\\text{VERIFY}\end{matrix}\right\}$ OPTION SPECIFIED. This error message alerts the user that either the FOLDing or VERify option was incorrectly specified.
UTSS9158	//DEF CARD WITH ID=PRT MUST BE SUPPLIED.

ERROR CODE	MESSAGE TEXT
UTSS9164	TERMINATED BY OPERATOR. CARTRIDGE WAS NOT MOUNTED. As the result of a reply of 'NO' to the message 'MOUNT PRINTER ARRANGEMENT NNNN ON DEVICE XXX. IS ARRANGEMENT MOUNTED?' and another 'NO' reply to the message 'IS RETRY DESIRED?'
UTSS9178	ABNORMAL RETURN FROM OPEN ON ID=$\begin{Bmatrix}\text{CARD}\\\text{PRT}\end{Bmatrix}$ RETURN INFORMATION=xxnn. See the Data Management message where nn are the two rightmost hex status completion code digits.
UTSS9188	MEMBER NNNN WAS NOT FOUND IN LIBRARY-ID=LIB. The first positional subparameter was not 'CARD' or one of the standard names, and was assumed to be the assembled and link-edited member of the library whose ID=LIB. The member was not found in this library.
UTSS9198	INVALID DEVICE ADDRESS SPECIFIED. This error message is the result of the 'LOADUCS' service request's unsuccessful search of the unit table, attempting to find an entry for the requested device.
UTSS9208	DEVICE TYPE IS NOT A SUPPORTED PRINTER. Printers supported by this utility are the IBM 1403 printers. The printer defined on the //DEF statement with ID=PRT was not a 1403 printer. If the printer was truly a 1403, then the 'Device Type' field of the unit table is invalid.
UTSS9218	BLANK OR NULL FOUND IN IMAGE. Either a blank (X'40' (no punches)) or a Null (X'00'(12-0-1-8-9 punch)) was found in the 240 byte image buffer prior to image load.
UTSS9228	IMAGE MEMBER NOT FOUND IN $NUCLIB. A UCS Load request was made for one of the standard images or a user supplied image corresponding to one of the names 'UC01' through 'UC10'. The 'LODNAM' request could not locate the member.
UTSS9238	$\begin{Bmatrix}\text{LESS}\\\text{MORE}\end{Bmatrix}$ THAN 4 DATA RECORDS HAVE BEEN FOUND. The user requested that his image be input from 'CARD'. Exactly four (4) non-comment cards must be provided.
UTSS9248	ABNORMAL RETURN FROM UCS LOAD REQUEST. RETURN INFORMATION=xxnn. For any abnormal return from the physical I/O request to execute a UCS Load command, the above message is displayed. xxnn is the return information from the I/O request. See the I/O Control message where nn are the two rightmost hex status completion code digits.
UTSS9258	PRINTER/CONTROLLER NOT EQUIPPED WITH UCS FEATURE. If the 'LOADUCS' service request attempts to load the image buffer of a printer designated (via unit table) as an IBM 1403, but receives command reject from the control unit, the above message is displayed.

F. EBCDIC BIT CONFIGURATIONS

Table F-1. EBCDIC Bit Configurations

	EBCDIC			EBCDIC	
		Bit Patterns			Bit Patterns
Graphic	Hole Patterns	Bit Positions 01234567	Graphic	Hole Patterns	Bit Positions 01234567
A	12-1	11000001	a	12-0-1	10000001
B	12-2	0010	b	12-0-2	0010
C	12-3	0011	c	12-0-3	0011
D	12-4	0100	d	12-0-4	0100
E	12-5	0101	e	12-0-5	0101
F	12-6	0110	f	12-0-6	0110
G	12-7	0111	g	12-0-7	0111
H	12-8	1000	h	12-0-8	1000
I	12-9	1001	i	12-0-9	1001
J	11-1	11010001	j	12-11-1	10010001
K	11-2	0010	k	12-11-2	0010
L	11-3	0011	l	12-11-3	0011
M	11-4	0100	m	12-11-4	0100
N	11-5	0101	n	12-11-5	0101
O	11-6	0110	o	12-11-6	0110
P	11-7	0111	p	12-11-7	0111
Q	11-8	1000	q	12-11-8	1000
R	11-9	1001	r	12-11-9	1001
S	0-2	11100010	s	11-0-2	10100010
T	0-3	0011	t	11-0-3	0011
U	0-4	0100	u	11-0-4	0100
V	0-5	0101	v	11-0-5	0101
W	0-6	0110	w	11-0-6	0110
X	0-7	0111	x	11-0-7	0111
Y	0-8	1000	y	11-0-8	1000
Z	0-9	1001	z	11-0-9	1001
0	0	11110000	Space	No punches	01000000
1	1	0001	¢ Cent Sign	12-8-2	1010
2	2	0010	. Period, Decimal Point	12-8-3	1011
3	3	0011			
4	4	0100	< Less Than Sign	12-8-4	1100
5	5	0101	(Left Parenthesis	12-8-5	1101
6	6	0110	+ Plus Sign	12-8-6	1110
7	7	0111	| Absolute	12-8-7	1111
8	8	1000			
9	9	1001			

Table F-1. EBCDIC Bit Configurations (Continued)

	EBCDIC			EBCDIC	
		Bit Patterns			Bit Patterns
Graphic	Hole Patterns	Bit Positions 01234567	Graphic	Hole Patterns	Bit Positions 01234567
& Ampersand	12	01010000	: Colon	8-2	01111010
! Exclamation Point	11-8-2	1010	# Number Sign	8-3	1011
$ Dollar Sign	11-8-3	1011	@ At Sign	8-4	1100
* Asterisk	11-8-4	1100	' Prime, Apostrophe	8-5	1101
) Right Parenthesis	11-8-5	1101			
; Semicolon	11-8-6	1110	= Equal Sign	8-6	1110
¬ Logical NOT	11-8-7	1111	" Quotation Mark	8-7	1111
- Minus Sign, Hyphen	11	01100000	⑀ Hook	12-0-9-8-4	11001100
			⑂ Fork	12-0-9-8-6	11001110
/ Slash	0-1	0001	⑁ Chair	11-0-9-8-4	11101100
, Comma	0-8-3	1011			
% Percent Sign	0-8-4	1100			
— Underscore	0-8-5	1101			
> Greater Than Sign	0-8-6	1110			
? Question Mark	0-8-7	1111			

G. UNIVERSAL CHARACTER SET IMAGES

AN Arrangement A, standard EBCDIC character set — 48 characters.

```
1234567890#@/STUVWXYZ&,%JKLMNOPQR-$*ABCDEFGHI+.□1234567890#@/STUVWXYZ&,%JKLMNOPQR-$*ABCDEFGHI+.□1234567890#@/STUVWXYZ&,%
JKLMNOPQR-$*ABCDEFGHI+.□1234567890#@/STUVWXYZ&,%JKLMNOPQR-$*ABCDEFGHI+.□1234567890#@/STUVWXYZ&,%JKLMNOPQR-$*ABCDEFGHI+.□
```

HN Arrangement H, EBCDIC character set for FORTRAN and COBOL, 48 characters.

```
1234567890='/STUVWXYZ&,(JKLMNOPQR-$*ABCDEFGHI+.)1234567890='/STUVWXYZ&.(JKLMNOPQR-$*ABCDEFGHI+.)1234567890='/STUVWXYZ&,(
JKLMNOPQR-$*ABCDEFGHI+.)1234567890='/STUVWXYZ&.(JKLMNOPQR-$*ABCDEFGHI+.)1234567890='/STUVWXYZ&.(JKLMNOPQR-$*ABCDEFGHI+.)
```

PCAN Preferred alphameric character set, arrangement A.

```
1234567890,-PQR#$@/STUVWXYZ□.*1234567890,-JKLMNOABCDEFGHI+.*1234567890,-PQR&$%/STUVWXYZ□.*1234567890,-JKLMNOABCDEFGHI+.*
1234567890,-PQR#$@/STUVWXYZ□.*1234567890,-JKLMNOABCDEFGHI+.*1234567890,-PQR&$%/STUVWXYZ□.*1234567890,-JKLMNOABCDEFGHI+.*
```

PCHN Preferred alphameric character set, arrangement H.

```
1234567890,-PQR=$'/STUVWXYZ).*1234567890,-JKLMNOABCDEFGHI+.*1234567890,-PQR&$(/STUVWXYZ).*1234567890,-JKLMNOABCDEFGHI+.*
1234567890,-PQR=$'/STUVWXYZ).*1234567890,-JKLMNOABCDEFGHI+.*1234567890,-PQR&$(/STUVWXYZ).*1234567890,-JKLMNOABCDEFGHI+.*
```

PN PL/1 alphameric character set.

```
1234567890XY/STUVW|:_",=JKLMNOPQR-Z(ABCDEFGHI+.)%$*#&@<;¬'?>1234567890XY/STUVW|:_",=JKLMNOPQR-Z(ABCDEFGHI+.)%$*#&@<;¬'?>
1234567890XY/STUVW|:_",=JKLMNOPQR-Z(ABCDEFGHI+.)%$*#&@<;¬'?>1234567890XY/STUVW|:_",=JKLMNOPQR-Z(ABCDEFGHI+.)%$*#&@<;¬'?>
```

QN PL/1 preferred alphameric character set for scientific applications.

```
1234567890XY/STUVW_"$*,=JKLMNOPQR-Z(ABCDEFGHI+.)1234567890XY/STUVW<;#*,=JKLMNOPQR-Z(ABCDEFGHI+.)1234567890XY/STUVW?>@*,=
JKLMNOPQR-Z(ABCDEFGHI+.)1234567890XY/STUVW¬'&*,=JKLMNOPQR-Z(ABCDEFGHI+.)1234567890XY/STUVW|:%*,=JKLMNOPQR-Z(ABCDEFGHI+.)
```

QNC PL/1 preferred alphameric character set for commercial applications.

```
1234567890#@/STUVWXYZ&.%JKLMNOPQR-$*ABCDEFGHI",=1234567890#@/STUVWXYZ&.%JKLMNOPQR-$*ABCDEFGHI|:_1234567890#@/STUVWXYZ&.%
JKLMNOPQR-$*ABCDEFGHI<;¬1234567890#@/STUVWXYZ$.%JKLMNOPQR-$*ABCDEFGHI'?>1234567890#@/STUVWXYZ&.%JKLMNOPQR-$*ABCDEFGHI)+(
```

Figure G-1. UCS Images

RN Preferred character set for applications of FORTRAN and COBOL.

1234567890XY/STUVW'@$*,=JKLMNOPQR-Z(ABCDEFGHI+.)1234567890XY/STUVW%@$*,=JKLMNOPQR-Z(ABCDEFGHI+.)1234567890XY/STUVW#@$*,=
JKLMNOPQR-Z(ABCDEFGHI+.)1234567890XY/STUVW□@$*,=JKLMNOPQR-Z(ABCDEFGHI+.)1234567890XY/STUVW&@$*,=JKLMNOPQR-Z(ABCDEFGHI+.)

SN Preferred character set for text printing.

1234567890&./STUVWXYZ,$*JKLMNOPQR-":ABCDEFGHI+abcdefghijklmnopqrstuvwxyz@'()–|1234567890&./STUVWXYZ,$*JKLMNOPQR-":ABCDEF
GHI+abcdefghijklmnopqrstuvwxyz@'()–| 1234567890&./STUVWXYZ,$*JKLMNOPQR-":ABCDEFGHI+abcdefghijklmnopqrstuvwxyz@'()–?!;¢ % □

TN Character set for text printing, 120 characters.

1234567890=./STUVWXYZ,# &JKLMNOPQR-":ABCDEFGHI+abcdefghijklmnopqrstuvwxyz@'?;±!$*% □ $^{1\,2\,3\,4\,5\,6\,7\,8\,9\,-+()0}$ ¬ ( ) ≠ + > < ≤ ≥ ¢ ■ [ ] }{ └┘ ┌┐● _ — |
1234567890=./STUVWXYZ,# &JKLMNOPQR-":ABCDEFGHI+abcdefghijklmnopqrstuvwxyz@'?;±!$*% □ $^{1\,2\,3\,4\,5\,6\,7\,8\,9\,-+()0}$ ¬ ( ) ≠ + > < ≤ ≥ ¢ ■ [ ] }{ └┘ ┌┐● _ — |

XN High-speed alphameric character set for 1403, Model 2.

1234567890,.ABCDEFGHIZJKLMNOPQR*$STUVWXY1234567890,.ABCDEFGHIZJKLMNOPQR*$STUVWXY1234567890,.ABCDEFGHIZJKLMNOPQR*$STUVWXY
1234567890,.ABCDEFGHIZJKLMNOPQR*$STUVWXY1234567890,.ABCDEFGHIZJKLMNOPQR*$STUVWXY1234567890,.ABCDEFGHIZJKLMNOPQR*$STUVWXY

YN High-speed preferred alphameric character set for 1403, Model 3 or N1.

1234567890STABCDEFGHIJKLMNOPQRUVWXYZ*,.1234567890STABCDEFGHIJKLMNOPQRUVWXYZ*,.#-$1234567890STABCDEFGHIJKLMNOPQRUVWXYZ*,.
1234567890STABCDEFGHIJKLMNOPQRUVWXYZ*,.1234567890STABCDEFGHIJKLMNOPQRUVWXYZ*,.#-$1234567890STABCDEFGHIJKLMNOPQRUVWXYZ*,.

Figure G-1. UCS Images (Continued)

H. LOAD UNIVERSAL CHARACTER SET (LOADUCS) SERVICE REQUEST

LOADUCS INSTRUCTION FORMAT

LOADUCS is the service request that loads any standard universal character set image or a user provided image. It is also used by Job Monitor and Spooler when a UCS is mounted. The format of LOADUCS is:

Name	Operation	Operand
label	LOADUCS	$\text{DEV=xxx,}\left[\begin{Bmatrix}\text{UCS=yyyy}\\ \text{DATABUF=zzzz}\end{Bmatrix}\right]\left[\text{,FOLD=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{YES}\end{Bmatrix}\right]$ $\left[\text{,PRINT=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{YES}\end{Bmatrix}\right]\left[\text{,STATUS=}\begin{Bmatrix}\underline{\text{NO}}\\ \text{YES}\end{Bmatrix}\right]$

Where:

label
Is an optional 1-8 character alphanumeric name; the first character must be an alphabetic.

DEV=xxx
Is the address (in hexadecimal) of the device that is to have its UCS buffer changed.

UCS=yyyy
Defines a standard character set (AN, HN, PCAN, PCHN, PN, QNC, QN, RN, SN, TN, XN, or YN) or a user provided image (UC01-UC10). A required parameter if DATBUF is not specified.

DATBUF=zzzz
A 240 byte user supplied image buffer. A required parameter if UCS is not specified.

FOLD=YES
Specifies folding on input data. The controller excludes bits 0 and 1 of all data bytes when comparing data bytes with image buffer characters. Default is NO.

PRINT=YES
Specifies that the 240 byte buffer is to be printed after image load. Default is NO.

STATUS=YES
Specifies a check of the printer for the UCS feature. Keywords DATBUF, PRINT, and FOLD are ignored. Default is NO.

LOADUCS MACRO PARAMETER PACKET

LOADUCS is a Class 5 service request. (See the **MRX/OS Control Program and Data Management Services, Extended Reference** manual.) It occupies either 5 or 7 words of storage; 5 if DATBUF is specified and 7 if DATBUF is not specified. Figure H-1 shows the LOADUCS macro expansion format.

	0 7	8 15
0	Op Code	Length
2	Return Information	
4	Flags	Tag
6	Pointer to Image Buffer	
8	Printer Device Address	
10	Image Name	
12		

Figure H-1. LOADUCS Macro Expansion

Byte	Bit	Description
0	0-7	The op code is $5B_{16}$.
	8-15	Length – the number of bytes in the parameter packet.
2		Return information.
4	0	1 if FOLD = YES 0 if FOLD = NO
	1	1 if PRINT = YES 0 if PRINT = NO
	2	1 if STATUS = YES 0 if STATUS = NO
6		Pointer to image buffer
8		Printer device address
10-12		Image name

INDEX

COMMENTS FORM

MRX/OS Utility Programs Reference Manual (2200.006-01)

Please send us your comments, to help us produce better publications. Use the space below to qualify your responses to the following questions, if you wish, or to comment on other aspects of the publication. Please use specific page and paragraph/line references where appropriate. All comments become the property of the Memorex Corporation.

- Is the material:

	Yes	No
Easy to understand?	□	□
Conveniently organized?	□	□
Complete?	□	□
Well illustrated?	□	□
Accurate?	□	□
Suitable for its intended audience?	□	□
Adequately indexed?	□	□

- For what purpose did you use this publication? (reference, general interest, etc.)

  ____

- Please state your department's function: ______________________________

  ____

- Please check specific criticism(s), give page number(s), and explain below:

  □ Clarification on page(s) ______________________________

  □ Addition on page(s) ______________________________

  □ Deletion on page(s) ______________________________

  □ Error on page(s) ______________________________

Thank you for your information. . . . .

Our goal is to provide better, more useful manuals, and your comments will help us to do so.

. . . . .Memorex Publications