4. TOSS UTILITIES

4.1 Introduction

The TOSS utilities are a set of routines which carry out various housekeeping functions on data files and program files. The routines are as follows:

Name	*Function*
CDD	Copy disk to disk
CFF	Copy file to file
CFT	Copy file to tape
CPP	Copy program
CRF	Create file
CRV	Create volume
CTF	Copy tape to file
DLF	Delete file
LFD	Load flexible disk
PDS	Print disk sector
PRF	Print file
PVC	Print volume table of contents
SCT	Scan tape
UDS	Update disk sector
UFD	Unload flexible disk.

All of the above routines except CPP operate on data files. CPP operates on disk resident program files.

TOSS utilities can be executed as independent programs or they can be link edited into a normal application program. The function of the present chapter is to describe the application program interface to TOSS utilities. The use of TOSS utilities as independent programs will not be described.

4.2 Copy Tape to File and Copy File to Tape

The utilities CTF and CFT are designed to handle ½" magnetic tape and cassettes labelled according to the ECMA–41 basic or compact labelling system. Multi-volume files are supported, and the different tracks of a cassette are regarded as different volumes.

4.3. Print File and Scan Tape

The following output formats are available with these utilities (PRF and SCT):

- Compact format

The record is printed with no conversion except for unprintable characters, which are replaced with ' . '. If unprintable characters have occurred '%%%' is output immediately after the contents of the record.

- Spaced format

Data is printed with spaces separating the characters. Unprintable characters are output as '#' followed by two hexadecimal digits representing the internal value. Space is printed as underscore, and underscore as $UN.

- Hexadecimal format

Data is printed in hexadecimal form with 16 bytes/line. The character equivalent is output in the right margin, with periods representing unprintable characters.

4.4. General Interface Rules

The calling sequence for all utilities is as follows:

```
LDKL    A12, pb-address
CF      A14, name
```

Where "pb-address" points to a parameter block which contains all the information needed by the system to run the utility, and "name" is the name of the utility (e.g. CDD, CFF). No validity check is performed by the utilities on the contents of the parameter block.

The utilities write a return code into register A1 before returning to the application program. If A1 is zero then no error has occurred. The significance of non-zero return codes is described in the utility reference section (4.5).

For some utilities register A2 contains an additional return code concerning I/O errors. This is the return code generated by the relevant I/O driver. The significance of these return codes is discussed in the driver reference section (3.4).

4.5 Utility Reference

This section contains detailed reference information for each TOSS utility.
The information for each utility is given in alphabetical utility name order. That is, first utility BIX, then CCF, and so on.

BIX BUILD INDEX FILE BIX

Description : This utility is used to create a sequential file that contains the index records for a given data file, by reading the data file in conjunction with parameters supplied by the programmer.

The input file is read until End-of-Medium is reached, and records containing no data are skipped. Both files must reside on disk, and the file to be created by the utility must have been previously defined with the CRF utility.

The record length of the output file must be set to the length of the key field + 6 characters.

Note: the output file from BIX is not ready to be used as an index file, but must first be sorted using the SRT utility, and then be input to the RIX (Reorganize Index) utility, in order to create the actual index files.

Calling Sequence :

```
LDKL  A12,pb-address
CF    A14,BINDEX
```

Parameter block :

word		
0	BUF1	
2	BUF2	
4		
6	FC3	FC4
8		
10		
12		
14		
16		
18		
20	DEV1	DEV2
22	KA	
24	KL	

BUF1	:	Data file sector buffer address
BUF2	:	Index (output) file sector buffer address
FC3	:	File code of input file
FC4	:	File code of output file
DEV1	:	Input device indicator; must be set to D
DEV2	:	Output device indicator; must be set to D
KA	:	Relative character address of key in data record; position 0 indicates 1st byte in record
KL	:	Key length in characters (1–255)

Continued

Return Code :	A1 = 0	No error
	A1 = 1	Input I/O error (A2 = return code)
	A1 = 2	Output I/O error (A2 = return code)
	A1 = 3	Blocksize error
	A1 = 4	Not used
	A1 = 5	Key definition is outside record boundaries
	A1 = 6	File organization is not S
	A1 = 7	Record length error
	A1 = 8	Output file is not large enough
	A1 = 9	Output file is not empty

CCF COPY CARDS TO DISK FILE CCF

Description : This utility is used to copy cards to a disk file previously defined using the CRF utility. The output file must be empty, otherwise an error message is output.

Reading and writing will be carried out until a card is read containing :EOF in columns 1–4. A message is then output to the operator's console stating how many cards have been copied.

If the output record length is greater than 80 bytes, the remainder of the records will be padded with blanks. If the record length of the output file is less than 80 bytes, the cards must contain blanks in columns having a number higher than the output record length, otherwise copying is terminated and an error message displayed on the operator's console.

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,COPYCF
```

Parameter block : word

0	BUF1	
2	BUF2	
4	╳	FC2
6	FC3	FC4

BUF1 : Address of record buffer 1
BUF2 : Address of record buffer 2
FC2 : File code of the operator's output device
FC3 : File code of the card reader
FC4 : File code of the disk output

Return code :

A1 = 0 No error
A1 = 1 Input I/O error (A2 = return code)
A1 = 2 Output I/O error (A2 = return code)
A1 = 3 Card reader not operable
A1 = 4 Output file not large enough
A1 = 5 Output file not empty
A1 = 6 Not used
A1 = 7 Faulty record length
A1 = 8 Disk not operable

CDD

COPY DISK TO DISK

CDD

Description : This utility is used to copy a whole disk to an empty disk already formatted.

The program makes a check that the output disk is empty. A check is also made that the VTOC of the output disk is large enough and that the free space on the output disk is sufficient.

The volume name is not copied to the output disk.

If a file on the input disk is divided into more than one extent, it will, if possible, be put together in one extent on the output disk.

Empty areas and badspots are not copied.

If some sectors cannot be copied due to a disk error the destroyed record numbers will be communicated to the operator.

Calling sequence :

```
LDKL  A12, pb-address
CF    A14,COPVOL
```

Parameter block :

word		
0	BUF1	
2	BUF2	
4	☒	FC2
6	FC3	FC4
8	Volume name 1	
10		
12		
14	Volume name 2	
16		
18		

BUF1 : The address of sector buffer 1
BUF2 : The address of sector buffer 2

FC2 : File code of the print device
FC3 : File code of the input disk
FC4 : File code of the output disk
Volume name 1 : Volume name of the input disk
Volume name 2 : Volume name of the output disk

CDD **Continued** **CDD**

Return code : A1 = 0 No error
A1 = 1 Input I/O error (A2 = driver return code)
A1 = 2 Output I/O error (A2 = driver return code)
A1 = 3 Volume name and file code do not correspond
A1 = 4 Output disk not empty
A1 = 5 VTOC overflow
A1 = 6 Disk overflow
A1 = 7 Faulty disk format
A1 = 8 Flexible disk write protected

CFF COPY FILE TO FILE CFF

Description : This utility is used to copy a disk file to an empty file area previously reserved using the CRF utility.

The program can be run with or without packing. If packing is used, records containing no data on the input file will not be copied. This means that randomly written files must be copied without packing.

Records placed after the last record number on the input file will also be copied.

The program checks that the last record number is zero on the output file and that the record lengths correspond on the input and output files.

The program also checks that the input and output file organizations are the same.

Calling sequence :

```
LDKL  A12, pb-address
CF    A14,COPFIL
```

Parameter block :

word		
0	BUF1	
2	BUF2	
4	╳	
6	FC3	FC4
8	╳	PACK

BUF1	:	The address of user buffer 1
BUF2	:	The address of user buffer 2
FC3	:	File code of the input file
FC4	:	File code of the output file
PACK	:	Pack indicator. Zero means no pack, non-zero means with packing.

Return code :

A1 = 0	No error
A1 = 1	Disk not operable
A1 = 2	Disk I/O error (A2 = driver return code)
A1 = 3	Faulty record length
A1 = 4	Output file not large enough
A1 = 5	Output file not empty
A1 = 6	File organization error.

CFT COPY FILE TO TAPE CFT

Description : This utility copies a disk file onto tape cassette or ½" magnetic tape.
All records up to end of file are copied, except for deleted (no data) records. Cassettes and magnetic tapes are written with or without labels. If labels are written, the following fields in the label record are copied from the VTOC:

- file name
- creation date
- retention period

If desired, several records can be blocked to form one block on tape.

The cassette or magnetic tape is not rewound at exit: it is therefore possible to copy several files from disk to the same cassette or tape. To rewind the cassette or tape the utility ULD (Unload Device) must be used.

Calling sequence :

```
LDKL  A12, pb-address
CF    A14,COPYFT
```

Parameter block : word

0	BUF1	
2	BUF2	
4	FC1	FC2
6	FC3	FC4
8		LBL2
10		BLF2
12		
14		SEQ2
16		
18	VOLFUN	
20		DEV2
22		
30	VOLID	
38		

CFT Continued CFT

BUF1 : The address of a buffer large enough to contain one tape block.
BUF2 : The address of a buffer at least 128 words long.
FC1 : Operators input device.
FC2 : Operators output device.
FC3 : File code of the input file.
FC4 : File code of the output file
LBL2 : Labelled tape flag. This must be 1 if the tape is labelled otherwise 0.
BLF2 : Block factor on output tape (binary value). This must be 1 if no blocking is required.
SEQ2 : Sequence number flag. If = 1, block sequence numbers will be written on to the output tape.
VOLFUN : Subroutine entry address.
When EOT on the output tape is detected, this routine is called (with a CFI A14, 18, A12 instruction), to request mounting of a new volume. Upon return FC4, and eventually VOLID, are assumed to be filled in with appropriate values.
DEV2 : Output device type - one ISO–7 character:
'T' if tape cassette, 'M' if ½" magnetic tape.
VOLID: The new tape volume identifier.
Only significant if LBL2 = 1.

Return code : A1 = 0 No error
A1 = 1 Output I/O error (A2 = driver return code)
A1 = 2 Input I/O error (A2 = driver return code)
A1 = 3 Erroneous blocksize - the block factor specified is too big. The blocksize becomes larger than allowed for the output device.

CIT COPY IBM FILE TO TOSS CIT

Description : This utility is used to copy a file from an IBM-labelled flexible disk to an empty file on a TOSS-labelled disk, previously created with the CRF utility.

The program can be run with or without packing. If packing is used, records indicating deleted data and records placed after the end of data on the input file will not be copied.

Data on the IBM-labelled disk must be EBCDIC-coded, and will be converted to ASCII code by the program.

The program checks that the LRN on the output file is zero, and that the input and output files have the same record length.

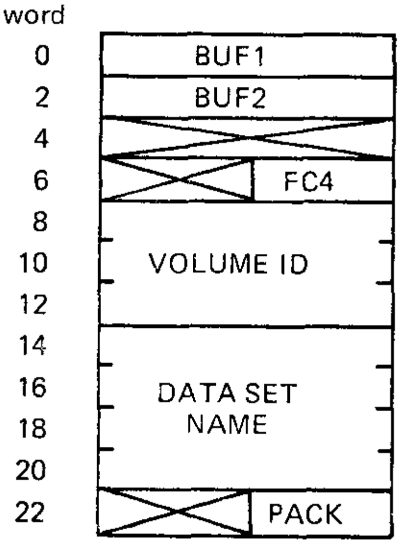
File codes /F8–/FB are used when searching for the volume identity. Sectors 08–26 are used when searching for the data set name.

Multivolume data sets are not allowed.

Calling sequence :

```
LDKL    A12,pb-address
CF      A14,COPYIT
```

Parameter block :



BUF1	:	Address of record buffer 1.
BUF2	:	Address of record buffer 2.
FC4	:	File code of output file.
VOLUME ID	:	Volume name of input disk.
DATA SET NAME	:	Name of input data set.
PACK	:	Pack indicator. Zero indicates no packing.

CIT **Continued** CIT

Return code	: A1 = 0	No error
	A1 = 1	Disk not operable
	A1 = 2	Disk I/O error (A2 = return code)
	A1 = 3	Faulty record length
	A1 = 4	Output file not large enough
	A1 = 5	Output file not empty
	A1 = 6	Data set name not found
	A1 = 7	Input volume name unknown

CPP COPY PROGRAM CPP

Description : This utility is used to move a program from a DOS6810 formatted volume and place it on a TOSS formatted volume.

A special UF-file on the DOS volume must contain the names and userids of the load modules in order to build up a library file. The sequence in which the module names have been generated in this file determines the segment numbers to be used when running the application. The "root" is segment 0 and the other ones are numbered from 1.

While creating the library file the load segment block table (LSBT) is built up in memory and, after the last segment of the file is written on disk, this table is stored in the first sector of the file.

As a result of the CPP utility there is a print out, on the operators typewriter, of all modules and the corresponding segment numbers, with segment lengths in bytes (hexadecimal).

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,COPPRO
```

CPP **Continued** CPP

Parameter block : word

word		
0	BUF1	
2	BUF2	
4	FC1	FC2
6	FC3	FC4
8	INPUT VOLUMENAME 1	
10		
12		
14		
16	OUTPUT VOLUMENAME 2	
18		
20		
22	INPUT FILENAME 1	
24		
26		
28	INPUT USERID	
30		
32		
34		
36	OUTPUT FILENAME 2	
38		
40		
42		

BUF1 : The address of the big sector buffer - 1640 words.

BUF2 : The address of the small sector buffer - 205 words.

FC1 : Filecode of the operators keyboard - /20.

FC2 : Filecode of the operators printer - /30.

FC3 : Filecode of the input disk (DOS).

FC3 : Filecode of the output disk (TOSS).

VOLUME NAME 1 : Volume name left adjusted and padded with spaces, written in the volume label of the input disk (DOS). No spaces are allowed in the volume name.

CPP Continued **CPP**

VOLUME NAME 2 : Volume name left adjusted and padded with spaces, written in the volume label of the output disk. (TOSS). No spaces are allowed in the volume name.

FILENAME 1 : File name left adjusted and padded with spaces. This is the name of the UF-file containing the module names and userids of modules to be copied into the file.

USERID : Userid left adjusted and padded with spaces. This is the name of the user where the UF-file resides (FILENAME 1). No spaces are allowed in the userid.

FILE NAME 2 : File name left adjusted and padded with spaces. This is the file where the library file is stored. No spaces are allowed in the file name.

Return code :	A1 = 0	No error
	A1 = 1	Disk not operable
	A1 = 2	Disk I/O error
	A1 = 3	Input filename unknown
	A1 = 4	Userid unknown
	A1 = 5	Input volume name unknown
	A1 = 6	Target area of output volume name too small
	A1 = 7	Output volume name unknown
	A1 = 8	Not used
	A1 = 9	Output filename unknown
	A1 = 10	Flexible disk write protected

CRF CREATE FILE CRF

Description : This utility is used to create a file on volumes already formatted.

The program searches for an empty space large enough to hold the file. If such an area is not available and a standard file is being created, it searches for a maximum number of 4 areas together being large enough to contain the file. The order in which the maximum of 4 volume names are given determines the order in which the volumes are searched. In the case of a library file only one file extent is allowed.

The volumes on which the file has to be created must be online together. File codes /F0–/FB are used when searching for the volume names. If free space has been found, the VTOC s of all volumes are updated. The file organisation in the VTOC is set to 'S' in the case of a standard file and to 'L' in the case of a library file. The created file will be filled with "FREE" records. These records will contain spaces.

Non-standard files will at this moment be treated in the same way as standard files.

On the operator's printer there is also a listing giving the volume name, file extent number, extent base and extent length for all file extents of the created file.

Calling sequence :

```
LDKL  A12, pb-address
CF    A14,CRFILE
```

CRF Continued CRF

Parameter block : word

word		
0	BUF1	
2	BUF2	
4		FC2
6	F.ORG	
8	FILE NAME	
10		
12		
14		
16	VOLUME-NAME 1	
18		
20		
22	VOLUME-NAME 2	
24		
26		
28	VOLUME-NAME 3	
30		
32		
34	VOLUME-NAME 4	
36		
38		
40	CREATION-DATE	
42		
44		
46	RETENTION PERIOD	
48		BF
50	RECORD LENGTH	

word	S-file		or	L-file	
52		KA		MONITOR NO	SOP SWITCH
54	NIF		or	PROGRAM LENGTH	
56	NO. OF RECORDS				

CRF **Continued** **CRF**

BUF1 : Address of the sector buffer.
BUF2 : Address of the print buffer.

FC2 : File code of the operator's print device.
F.ORG : File organisation copied to the VTOC for the created file.
FILENAME : File name left adjusted and padded with spaces, written in the VTOC for the created file. No spaces are allowed in the file name.

VOLUME NAME 1–4 : Names of volumes where a search for free space areas. Unused entries should contain spaces.
CREATION DATE : Six characters copied to the VTOC for the created file.
RETENTION PERIOD : Three characters copied to the VTOC for the created file.

BF : Blocking factor. That is the number of records within one block (one byte binary value). The BF is set to 1 if the file is a library file.

RECORD LENGTH : One word binary value. Note that the status character maintained on disk for each record is not included in the record length. The record length is set to 400 if this is a library file.

For S-files

NIF : One byte binary value, being the number of index files related to this data file.

KA : Binary value, indicating the relative character address of the symbolic key within the data record (0–n). This is only significant for index files and must otherwise be zero.

NO. OF RECORDS : 1 word binary value. Not relevant for L-files. Max. value = 8388607.

For L-files

MONITOR NO. : One byte binary value indicating which Monitor has to be used together with the application.

SOP SWITCH : One byte binary value indicating which switch on SOP-panel to be used at application loading time.

PROGRAM LENGTH : Program length in k bytes.

CRF **Continued** **CRF**

Return code :	A1 = 0	No error
	A1 = 1	Disk I/O error
	A1 = 2	One or more volumes unknown
	A1 = 3	Disk overflow
	A1 = 4	File could not be allocated within 4 extents.
	A1 = 5	Blocklength greater than 400 characters.
	A1 = 6	VTOC overflow
	A1 = 7	Filename already used
	A1 = 8	Flexible disk write protected.

CRV CREATE VOLUME CRV

Description : This utility is used to format a disk pack before it will be used. It writes a volume label (VL) and an empty volume table of contents (VTOC) on disk. The program also writes cylinder identifiers in all sectors on the disk and tests the quality of each sector. If an unusable sector is found, this sector is withdrawn from the user available area and a dummy file BADSPOT is created which occupies the sectors not to be used. File organization is set to "B". The free space administration table in the VTOC is also updated for the "badspot" area. If the disk is PTS6875 or PTS6876, the Initial Program Loader (IPL) is stored in Sector 1, Cylinder 0, Track 0; if flexible disk it is stored in physical sectors 5–8, Track 00.
It is possible to choose if the IPL shall read the application directly and start the system (only one application on disk) or first wait for SOP input to define the application that shall be read (more than one application on disk).

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,CRVOL
```

Parameter block :

word		
0	BUF1	
2	BUF2	
4		FC2
6		FC4
8	VOLUME	
10	NAME	
12		
14	VTOCR	
16	DTYP	APL

BUF1 : Address of sector buffer 1.
BUF2 : Address of sector buffer 2.

FC2	:	File code of the operator's print device.
FC4	:	File code of the disk to be formatted.
VOLUME NAME	:	Volume name left adjusted and padded with spaces written in the volume label of the disk to be formatted. No spaces are allowed in the volume name.
VTOCR	:	Number of empty records reserved in the volume table of contents (VTOC) (1 word binary value).

CRV **Continued** **CRV**

DTYP : Disk unit type indicator. Zero means PTS6875 or Flexible Disk, non-zero means PTS6876.

APL : Indicator for the number of applications to be stored on the disk. Zero means one application. Not zero means more than one application.

Return code : A1 = 0 No error
A1 = 1 Disk spot operable
A1 = 2 Bad spot on track zero. Disk not usable
A1 = 3 More than 5 bad spots. Disk not usable
A1 = 4 Disk I/O error (A2 = driver return code)
A1 = 5 Flexible disk write protected.

CTF COPY TAPE TO FILE CTF

Description : This utility copies from a cassette tape or ½" magnetic tape file to a disk file. The disk file is written with sequential write, so that if the output file is not empty, the new data will be appended. Cassettes can be read with or without labels, magnetic tapes only without labels.

The following fields in the label records will be checked:

- label identifier
- file section number
- volume identifier (only if specified)

Blocked records on input are accepted, and deblocking will be performed.

If input records are blocked, the blocksize must be a multiple of the record length for the output file, otherwise records with a different size will be truncated or expanded with blanks, to fit into the output file.

In the case of a labelled input tape, the check of the header label might fail for one of the following three reasons:

- erroneous label identifier: this is not a header label record.
- erroneous volume identifier: the volume identifier specified in the parameter block differs from the corresponding field in the header label record.
- file section sequence error.

In these cases a message and the label record will be displayed on the operators console.

The operator will then be requested to enter one of the following three commands:

GO	-	ignore the error
REMOUNT	-	another volume should be mounted
ABORT	-	terminate the copying

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,COPYTF
```

CTF Continued **CTF**

Parameter block :

word		
0	BUF1	
2	BUF2	
4	FC1	FC2
6	FC3	FC4
8	LBL1	
10	BLF1	
12		
14	SEQ1	
16	VOLFUN	
18		
20	DEV1	
22	VOLID	
30		

BUF1 : The address of a buffer, large enough to contain one tape block.

BUF2 : The address of a buffer, at least 128 words.

FC1 : Operator's input device.

FC2 : Operator's output device.

FC3 : File code for the input device.

FC4 : File code for the output device.

LBL1 : Labelled tape flag. This must be 1 if the input tape is labelled: 0 otherwise.

BLF1 : Block factor on input tape (binary value). This must be 1 if no blocking is used.

SEQ1 : Sequence number flag (binary value). If = 1, block sequence numbers are expected on the input tape.

VOLFUN : Subroutine address. When the end of volume label is read on tape, this routine is called (with a CF A14, ... instruction), to request mounting of the next volume. Upon return FC4 and - eventually - VOLID are assumed to be filled in with appropriate values.

CTF **Continued** **CTF**

DEV1 : Input device type - one ISO–7 character:
'T' if tape cassette, 'M' if ½" magnetic tape.

VOLID : Volume identifier of the input tape.
If it is not all spaces, this field is compared with the corresponding field in HDR-label.

Return code : A1 = 0 No error
A1 = 1 Output I/O error (A2 = driver return code)
A1 = 2 Input I/O error (A2 = driver return code)
A1 = 4 Tape mark missing
A1 = 5 Tape mark after label missing
A1 = 6 Label record missing
A1 = 7 Unexpected tape mark
A1 = 8 Label record error
A1 = 9 Input length error - only when blocked input. This means that the blocksize is not a multiple of the record length.
A1 = 10 End of File detected inside a split record.
A1 = 11 HDR label check,
A2 = 1 : Label identifier error — not HDR
A2 = 2 : Volume identifier check
A2 = 3 : File section sequence error.
A1 = 12 Output file not large enough.

CTI COPY TOSS FILE TO IBM DATA SET CTI

Description : This utility is used to copy a file from a TOSS-labelled disk to a data set on an IBM-labelled disk. It cannot be used to copy L-type files.

The program can be run with or without packing. If packing is used, records indicated as deleted in the input file will not be copied, and the output records will be written sequentially. This means that files which are used in the random read must be copied without packing.

Records placed after the EOF in the input file will be copied.

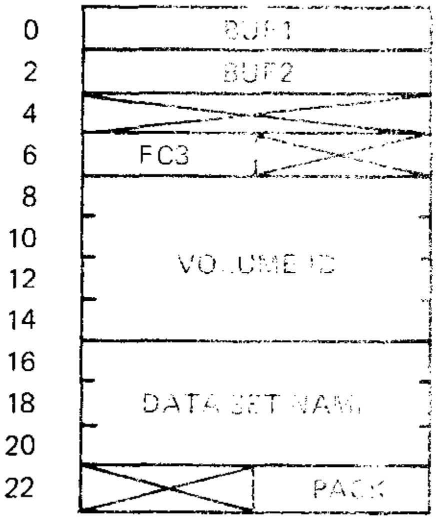
The input records must be in ASCII format and will be converted to EBCDIC format by the program.

Multivolume data sets are not allowed.

File codes /F8–/FB are used when searching for the output volume identity. Sectors [illegible] Track 00 are used when searching for the data set name.

Calling sequence :
```
LDKL  A12,pb-address
CF    A14,COPYTI
```

Parameter block : word



BUF1	:	Address of record buffer 1
BUF2	:	Address of record buffer 2
FC3	:	File code of input file
VOLUME ID	:	Volume name of output disk
DATA SET NAME	:	Name of output data set
PACK	:	Pack indicator. Zero indicates no packing

CTI **Continued** **CTI**

Return code :	A1 = 0	No error
	A1 = 1	Disk not operable
	A1 = 2	Disk I/O error
	A1 = 3	Faulty record length
	A1 = 4	Output file not large enough
	A1 = 5	Output file not empty
	A1 = 6	Data set name not found
	A1 = 7	Output volume name unknown
	A1 = 8	Flexible disk write protected

DLF | DELE[illegible] | DLF

Description : This utility [illegible] file containing a loadable program [illegible] file to be deleted resides on more [illegible] different volume names are entere[illegible]

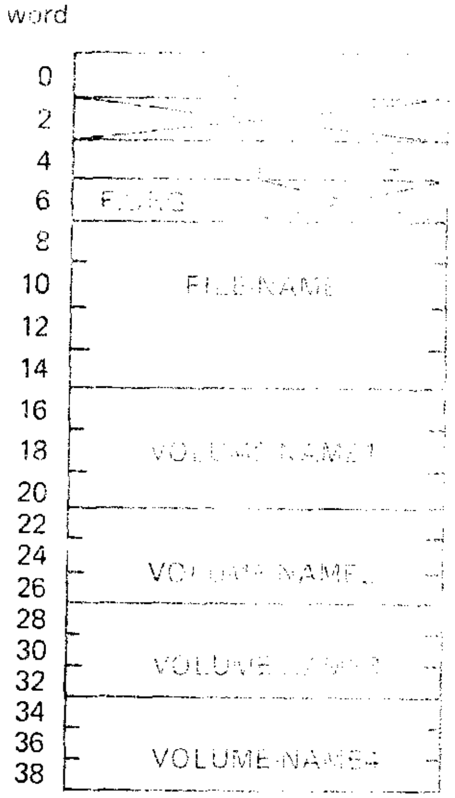
The free space admin[illegible] contained in VTOC is updated for the relea[illegible] file descriptor record in VTOC is set to [illegible] file name.

File codes /F0—/F[illegible] for the volume names.

Calling sequence :

```
LDKL  A12,[illegible]
CF    A14,DL[illegible]
```

Parameter block :



BUF1	:	Address of the sector buffer
F.ORG	:	File organisation.
FC2	:	File code of the operator's print device.
FILENAME	:	File name of the file to be deleted.
VOLUME NAME	:	Name of volumes where a search is made for the file to be deleted. Unused entries should contain spaces.

Continued

Return code	: A1 = 0	No error
	A1 = 1	Disk I/O error
	A1 = 2	Volume name unknown
	A1 = 3	File name unknown
	A1 = 4	No entry available in free space table of VTOC
	A1 = 5	Flexible disk write protected.

PDS PRINT DISK SECTOR PDS

Description : This utility will give a listing of the contents of one or more sectors on a specified disk. Output format is hexadecimal together with ISO–7 representation. 410 bytes per sector are printed, except in the case of an IBM-formatted flexible disk, in which case 128 bytes per sector are printed.

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,PRDISC
```

Parameter block :

word		
0	BUF1	
2	BUF2	
4		
6	FC3	FC4
8	SECTOR NUMBER 1	
10	SECTOR NUMBER 2	

BUF1 : Address of the disk sector buffer
BUF2 : Address of the print buffer
FC3 : File code of the print device
FC4 : File code of the disk to be printed
SECTOR NUMBER 1: Sector number of the first sector to be printed
SECTOR NUMBER 2: Sector number of the last sector to be printed.

Return code :

A1 = 0 No error
A1 = 1 Disk not operable
A1 = 2 Disk I/O error
A1 = 3 Printer not operable
A1 = 4 Illegal sector number

PIT PRINT INDEX TRACK PIT

Description : This utility is used to print a listing of all the data relevant to the index track of an IBM-formatted flexible disk, i.e. Track 00, Sectors 1–26.

For each data set defined on the volume, the following information is printed –

Data Set Label Sector
Data Set Name
Record Length
Beginning of Extent Track and Sector Number
End of Extent Track and Sector Number
Creation Date
Expiry Date
End of Data Track and Sector Number

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,PRINDX
```

Parameter block :

word		
0	BUF1	
2	BUF2	
4		FC2
6	FC3	FC4

BUF1	:	Address of disk sector buffer
BUF2	:	Address of print buffer
FC2	:	File code of operator's print device
FC3	:	File code of print device
FC4	:	File code of disk containing Index Track to be printed

Return code :

A1 = 0	No error
A1 = 1	Disk not operable
A1 = 2	Disk I/O error (A2 = Return Code)
A1 = 3	Printer not operable
A1 = 4	Disk is not IBM format

PRF PRINT FILE PRF

Description : This utility lists the records in a file on the line printer or operator's console.

Calling sequence :
```
LDKL  A12,pb-address
CF    A14, PRFILE
```

Parameter block :

word		
0	BUF1	
2		
4		
6	FC3	FC4
8	CNV	PRNR
10	LWB	
12	UPB	

BUF1 : Buffer address.
FC3 : File code for the input file.
FC4 : File code for the print device.
CNV : Output conversion:
= 0 – compact format
= 2 – spaced format
= 4 – hexadecimal format.
PRNR : Record number print flag
≠ 0 – the record number will be printed before data.
LWB : Number of the first record to be printed. } Bit 0 must be zero
UPB : Number of the last record to be printed. }

Return codes :
A1 = 0 No error
A1 = 1 Input error (A2 = driver return code)
A1 = 2 Disk I/O error (A2 = return code)
A1 = 3 Printer not operable
A1 = 4 End of medium (A2 = last record number in file)

PVC PRINT VOLUME TABLE OF CONTENTS PVC

Description : This utility will give a listing of all relevant data in the VTOC. For all datasets defined in the VTOC the following items will be printed.

- file name
- file extent base
- file extent length
- file organisation
- record length
- blocking factor
- last record number
- file section number
- file extent number
- creation date
- retention period
- number of index files
- symbolic key address in data record
- monitor number
- switch number on SOP

Number of free records in the VTOC and free space administration table are also printed.

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,PRVTOC
```

Parameter block :

word		
0	BUF1	
2	BUF2	
4		
6	FC3	FC4

BUF1 : Address of the disk sector buffer
BUF2 : Address of the print buffer
FC3 : File code of the print device
FC4 : File code of the disk containing the VTOC to be printed.

Return code :

A1 = 0	No error
A1 = 1	Disk not operable
A1 = 2	Disk I/O error (A2 = driver return code)
A1 = 3	Printer not operable
A1 = 4	Not TOSS disk format.

RIX REORGANIZE INDEX FILE RIX

Description : This utility is used to create an index file and a master index file from a sorted file previously created by the BIX utility. This file is read until End-of-medium.

All files used must reside on disk, and the output files must have been predefined with the CRF utility. The index file may not span more than one volume. The master index file must reside on the same volume as the index file, and must be contained within one extent. It must have a record length equal to RL–3, where RL is the record length of the index file.

The utility moves index records sequentially from the input file to the output file. Free records are inserted at the end of each sector according to the Load Factor parameter supplied by the program. The master index is also written sequentially.

A check is performed on the input record sequence, and the processing terminated if an error is detected.

Each record in the master index file corresponds to one master index entry. The maximum number of index records per master index entry is calculated according to the following formula:

$$NIM = \uparrow \frac{NORI}{FELM \times BFM}$$

where $\uparrow$ = Next higher multiple of Index file blocking factor
NORI = Number of records in Index file
FELM = Number of blocks in Master Index file
BFM = Blocking factor of Master Index file

The number of used index records per sector of the output index file is calculated from the Load Factor parameter, according to the following formula:

$$NUI = \downarrow BF \times \frac{LF}{100}$$

where $\downarrow$ = Next lower integer value
BF = Blocking factor of Index file
LF = Load factor

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,RINDEX
```

RIX **Continued** RIX

Parameter block : word

word		
0	BUF1	
2	BUF2	
4	FC1	FC2
6	FC3	FC4
8	FC5	
10		
12		
14		
16		
18		
20	DEV1	
22		
24	LF	
26		
28	NOR	

BUF1 : Input file sector buffer address

BUF2 : Output index file sector buffer address

FC1 : File code of operator's input device; set to zero if no operator-controlled abort is required.

FC2 : File code of operator's print device; set to zero if no print required

FC3 : File code of input file

FC4 : File code of output Index file

FC5 : File code of output Master Index file

DEV1 : Device indicator; must be set to D

LF : Load Factor for output index file, i.e. percentage of the sector filled with used records.

NOR : Number of reserved records in the output Index file. A binary value in three characters with bit 0 in the second character set to zero.

Return code :

A1 = 0	No error
A1 = 1	Input I/O error (A2 = return code)
A1 = 2	Output I/O error (A2 = return code)
A1 = 3	Blocksize error
A1 = 4	Not used
A1 = 5	Key definition error
A1 = 6	Load factor too small
A1 = 7	Record length error
A1 = 8	Output file not large enough
A1 = 9	Output file not empty
A1 = 10	Not used
A1 = 11	Key sequence error
A1 = 12	File organization error

SCT SCAN TAPE SCT

Description : This utility is a comprehensive tool for examining the contents of a tape cassette or ½" magnetic tape. The program interacts with the operator by means of a small set of commands for tape positioning and printout control.

On entry, the utility executes a load request for the input device. Then commands are requested, interpreted and executed, until an exit command is encountered, when an unload request is executed on the input device and control is returned to the calling program.

The following control commands are available:

R - rewind tape
F - forward positioning
B - backward positioning
L - list blocks on operator's console
P - list blocks on print device
O - change output format
X - return to calling program

The command syntax is defined as follows:

```
One command ::=   com// eoc   |   com //  parlist // eoc

com      ::= one character command-mnemonic
eoc      ::= end-of-command character
parlist  ::=  par          | npar // aparlist
                           | npar // sep // parlist
                           | apar // parlist
                           | sep // parlist
par      ::= apar |  npar
apar     ::= one-alphabetical-character
npar     ::= unsigned-decimal-number
aparlist ::= parlist -starting-with-an- apar
sep      ::= comma |space
```

The "command list" feature allows the operator to write several commands on the same line, and then have them executed in the order they were entered. A command list is a list of commands, separated by periods and followed by an end-of-record key. A command can be cancelled by pressing CR before the end-of-command character * is entered.

The commands are described in detail in the following paragraphs:

R - rewind tape
syntax: R // eoc
function: the tape is rewound to BOT

F - forward positioning
Syntax: F [F] [n_1 [,n_2]] eoc

SCT **Continued** SCT

F: If this parameter is given the positioning is performed on file (i.e. tape mark) level, otherwise on block level.

n_1: numeric parameter-number of blocks or, if the parameter F is present the number of files-to skip.

n_2: numeric - number of tapemarks after which the operation stops. Only significant if the parameter F is absent.

B - backward positioning

Syntax: B [F] [n_1 [, n_2]] eoc

parameters as for the command F.

L - list blocks on operator's console

Syntax: L: [n_1 [, n_2]]
n_1: number of blocks to list.
n_2: file-count - the operation will stop when n_2 tape marks have been encountered.

P - list blocks on print device

Syntax: P [n_1 [, n_2]]
parameters as for the command L.

O - change output format

Syntax: O [fmt] [n_1 [, n_2]]
fmt: one or more output format mnemonic characters, valid characters are:

A: ISO–7 code
E: EBCDIC code
C: compact format
S: spaced format
H: hexadecimal format
any other character will be ignored.
n_1: first position in block to output.
n_2: last position in block to output.
default for n_1 and n_2 : the contents of the whole block is printed. Character position count starts with 0.

X - exit from the utility

Syntax: X

Calling sequence : LDKL A12, pb-address
CF A14,SCANT

SCT Continued SCT

Parameter block :

0	BUF1	
2		
4	FC1	FC2
6	FC3	FC4
8	CNV	
10	CODE	

BUF1 : Address of the buffer
FC1 : File code for the operator's input device.
FC2 : File code for the operator's output device.
FC3 : File code for the input tape.
FC4 : File code for the print device.
CNV : Output conversion format code
0 = compact format
2 = spaced format
4 = hexadecimal format
CODE : Character code
0 = ISO–7
2 = EBCDIC

Return code : A1 = always zero.

SRT SORT FILE SRT

Description : This utility is used to sort records in a file into a sequence determined by key fields contained in the records. The key may consist of up to 15 subkeys, each subkey consisting of a number of eight-bit characters.

The utility requires two or three disk files. If only two files are used, the input file is destroyed, so it is important to keep a backup copy of the file until the sort has been successfully processed. If three files are used, the input file is unaffected.

The input and output files may have different record lengths, in which case records are truncated and/or padded with blanks, as appropriate. If the files have different record lengths, the sort keys must be contained within the shortest record length.

The work files and the output file must have been predefined with the CRF utility. All files must reside on disk. Files are handled with Random Access, and the LRN of the output file is not significant.

The utility also requires a work area in memory, which should be defined as large as possible to decrease the length of time required for sorting. The maximum size is 64 Kbyte. The minimum core size can be calculated by the following formula:

4 x RL x BFC

where RL = Effective record length (see parameter block layout)

BFC = Lowest common blocking factor of input, output and work files

A stable sort algorithm is used, which means that the initial relative order of equal keys in the input file is kept in the output file.

The sort executes in the following four phases:

Precalculation: Secondary sort parameters (number of passes, etc) are calculated and the results output to the operator's print device. The operator may abort the sort at this stage.

Presort: Used records are read in sequence from the input file to the memory area. Records in memory are sorted and written in sequence to a work file, forming a 'file division'. This phase is repeated until the entire input file is moved to the other file.

Merge: Records from separate file divisions on one file are merged and output to another file forming larger file divisions. Statistics are printed on the operator's print device. This phase is repeated until the output file contains only one file division.

Check: The output file is read in sequence and fifteen checks are carried out on it. Statistics are gathered and output on the operator's print device.

At the end of the sort processing, the TNR field in the parameter block (see below) is updated with the number of records read from the output file in the Check Phase, excluding free records.

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,SORTF
```

Continued

Parameter block : word

word		
0	☒	
2		
4	FC1	FC2
6	FC3	FC4
8	FC5	☒
10	☒	
12	PO	SO
14	☒	
16		
18	CONPAR	
20	DEV1	DEV2
22	CWA	
24	CWAL	
26	NDB	TNR
28	☒	
30	RL	
32		
34		
36		
38		
40	NSK	
42	KA1	
44	KA2	
46	KA3	
	|	
70	KA15	
72	NKC1	
74	NKC2	
76	NKC3	
	|	
100	NKC15	

Positions 32–39 are set to /00

SRT **Continued** **SRT**

FC1 : File code of operator's input device; set to zero if operator abort function is not required

FC2 : File code of operator's print device; set to zero if not print required

FC3 : File code of input file

FC4 : File code of output file

FC5 : File code of work file

Note: When FC3 and FC4 are the same, the output file will overwrite the input file.
When FC3 and FC5 are the same, the input file is used as a work file.

PO : Processing order; 0 = single task processing
1 = multitask processing

SO : Sorting order; 0 = descending sequence
1 = ascending sequence

CONPAR : Address of the linked-in object module parameter block used for this sort

DEV1 : Input device indicator; set to D

DEV2 : Output device indicator; set to D

CWA : Core work area address

CWAL : Size of the work area in memory in number of bytes; binary value <65536

NDB : Number of disk buffers available in the MONITOR. A binary value in the range 1–5. The higher the number, the shorter the sorting time.

TNR : Number of records that participate in the sort; both free and used records in the input file are read up to this number of records. If set to zero, the entire input file is used. Binary value in three characters; bit 0 in the second character must be set to zero

RL : Binary field indicating the requested effective record length. This is the number of leading characters in the input record that participate in the sort, and the sort key must be contained within this length. If set to zero the effective record length is the record length of the input file

NSK : Binary field indicating the number of subkeys, 1–15

KA1–15 : Binary fields indicating the relative character positions of the subkeys in the input records. KA1 refers to the most significant subkey. The first position in the record is assumed to be 0 for this field, and the maximum value is 255

NKC1–15 : Binary fields indicating the number of characters in each of subkeys KA1–15 respectively to a maximum value of 255

SRT **Continued** **SRT**

Return code :	A1 = 0	No error
	A1 = 1	Input I/O error (A2 = return code)
	A1 = 2	Output I/O error (A2 = return code)
	A1 = 3	Not used
	A1 = 4	Not used
	A1 = 5	Key definition error
	A1 = 6	Work area in memory too small
	A1 = 7	Record length error
	A1 = 8	Output file not large enough
	A1 = 9	File organization not S
	A1 = 10	Not used
	A1 = 11	Incorrect TNR (see parameter block)

UDS UPDATE DISK SECTOR UDS

Description : This utility is used to change one or more positions in a specified sector on disk. The utility may be used on IBM-formatted flexible disks.

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,UPDISC
```

Parameter block :

word		
0	BUF1	
2		
4		
6		FC4
8	SECTOR NUMBER	
10		
12	POS 1	
14	POS 2	
	HEX. VALUE MAX 24 POSITIONS	
36		
38		

BUF1 : Address of the disk sector buffer.
FC4 : File code of the disk to be updated.
SECTOR NUMBER : Sector number of the sector to be updated.
POS1 : Character position of the first character to be updated.
POS2 : Character position of the last character to be updated.
First position in the disk sector is POS = 0.
Note that position 0 and 1 cannot be changed as they are reserved for the System.
HEX VALUE : A string of up to 24 characters giving the hexadecimal representation of the positions to be changed.

Return code :

A1 = 0	No error
A1 = 1	Disk not operable
A1 = 2	Disk I/O error
A1 = 3	Flexible disk write protected
A1 = 4	Illegal sector number
A1 = 5	Illegal position number (only for IBM disk format).

WIL WRITE IBM LABELS WIL

Description : This utility is used for one of three purposes: to write labels on a flexible disk in order to prepare it for further use and create a data set at the same time, to subsequently create further data sets on the disk, or to delete data sets from the disk.

The program initialises track 00 on the disk by writing the volume identity into positions 5–10 of sector 07, and subsequently uses sectors 08–26 for each data set label. It is recommended that sector 08 is used when creating the first data set, and then 09,10, etc. in sequence. When creating further data sets, the program checks that the volume identity is correct and that the data set name does not already exist on the disk. The positions in each sector used to record data set label information are as follows:

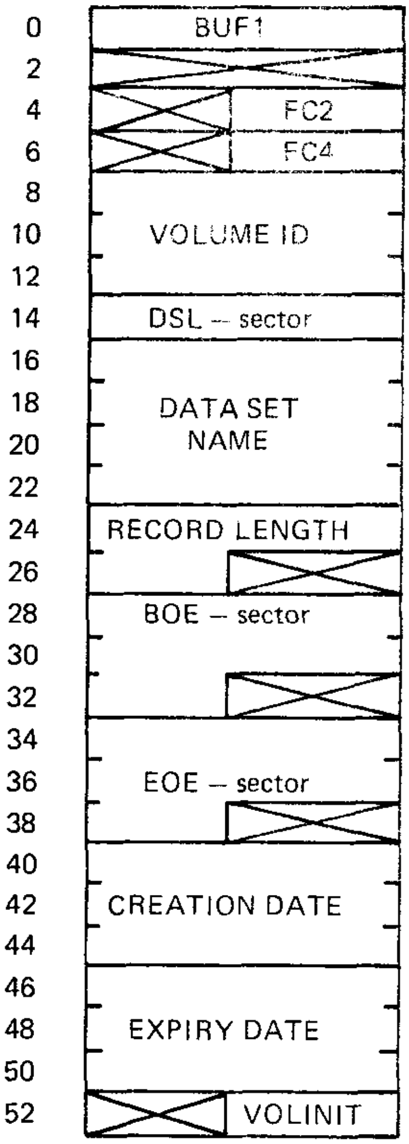
Position	6–13	User name for data set
	23–27	Record length
	29–33	Beginning of extent
	25–29	End of extent
	48–53	Creation date
	67–72	Expiry date
	75–79	End of data

Calling sequence :

```
LDKL  A12,pb-address
CF    A14,IBMLBL
```

WIL Continued **WIL**

Parameter block : word



BUF1 : Address of disk sector buffer

FC2 : File code of operator's print device

FC4 : File code of disk (/F8–/FB)

VOLUME ID : Volume identity – six alphanumeric characters, left-justified, and with trailing blanks if necessary.

DSL – sector : Binary field, containing the physical sector number of the data set label.

WIL **Continued** WIL

DATA SET NAME : User data set name (max 8 characters)

RECORD LENGTH : 3 digits specifying how much of each 128 position sector contains actual data.

BOE – sector : Physical sector of beginning of extent, in the format TT0SS, where
TT = track number
0 = zero
SS = sector number

EOE – sector : Physical sector of end of extent, in the same format as beginning of extent.
Note: To delete a data set from the disk, the BOE and EOE fields in the block must be set to 74001 and 73026.

CREATION DATE : Six digits in the format YYMMDD or blanks

EXPIRY DATE : Six digits in the format YYMMDD or blanks

VOLINIT : Volume initialisation indicator. Zero means initialisation is not required.

Return code :

A1 = 0	No error
A1 = 1	Disk not operable
A1 = 2	Disk I/O error (A2 = return code)
A1 = 3	Volume identity unknown
A1 = 4	Data set name already used
A1 = 5	Flexible disk write protected
A1 = 6	Not used
A1 = 7	Data set definition error