DSKRAT

DISK FILE DAMAGE ASSESSMENT PROGRAM

Date: 14-Jun-73
File: DSKRAT.RNO
Edition: 2

This document reflects the software of version 10.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation.

Actual distribution of the software described in this specification will be subject to terms and conditions to be announced at some future date by Digital Equipment Corporation.

DEC assumes no responsibility for the use or reliability of its software on equipment which is not supplied by DEC.

This software is furnished to purchaser under a license to use on a single computer system and can be copied (with inclusion of DEC's copyright notice) only for use in such system, except as may otherwise be provided in writing by DEC.

1.0 INTRODUCTION

DSKRAT is a damage assessment program for Level D disk file structures. It scans the file structure, using super USETI's to read files, and reports any inconsistencies detected to device LPT. File name RAT.LST is used if LPT is a directory device.

2.0 DSKRAT OPERATIONS

To run DSKRAT, log in as [1,2] (required for super USETI), assign to a file structure the logical name STR, assign LPT if you wish it to be some device other than the printer, and run DSKRAT. DSKRAT opens STR and LPT, reads the SAT blocks into core and, if everything is acceptable, types RUNNING, and begins to scan the file structure.

DSKRAT looks up each file on the file structure and reports any failures, reads and verifies all RIBs of the file, checksums each group and reports errors, reads the retrieval information from the RIBs and constructs its own SAT blocks. If there are any disagreements between the SAT's read from disk and SAT's constructed by DSKRAT, error messages are output. That is, if any cluster is in more than one file, or in a file but not marked in the SAT, that fact is reported and the cluster and the file to which it belongs are identified. One line is produced for each error. It includes the file name, cluster number and logical block number of the cluster in question, and an error comment.

When DSKRAT has gone through the file structure completely, it prints a list of clusters in more than one file, clusters in files by not marked in SAT's, and clusters marked in SAT's but not in any file. Then, if any clusters are in more than one file, it types

```
END OF PASS 1, BEGINNING PASS 2
```

and starts over. The second pass will produce an error line for every file claiming clusters used by more than one file. The first such file is not known in pass 1 until the second is found. If pass 2 is not wanted, type CONTROL-C twice and REENTER. This will close LPT and DSKRAT will exit.

If no clusters are in more than one file, DSKRAT types

```
END OF PASS 1, NO NEED FOR PASS 2
```

and exits.

Note that since DSKRAT reads in SAT's at the beginning of the program, if other users are referencing the disks (writing or deleting files or reading files marked for

deletion) the SAT blocks will not be current and spurious errors will appear. However, true errors will not be missed.

3.0 FLOW FOR DSKRAT

Read and verify home blocks, print if errors are found.

Save parameters.

Print the file structure name, blocks per cluster, time, and date.

Initialize disk SAT (find SAT.SYS RIB from the home block and read the SAT blocks).

Clear the computed SAT and the trouble SAT.

Read and verify MFD RIB, print if errors are found.

Type RUNNING.

3.1 PASS1:

Call DOMFD.

3.2 END PASS 1:

List all lost clusters. (For each cluster, list if the non-computed-SAT and the disk-SAT = 1)

List all free clusters. (For each cluster, list if the computed-SAT and the non-disk-SAT = 1)

List all clusters used for more than one file. (For each cluster, list if the trouble-SAT = 1)

Type END OF PASS 1

If there are no clusters used for more than one file, type

NO NEED FOR PASS 2

and exit.

Otherwise, type BEGINNING PASS 2

3.3 PASS 2: Clear the computed SAT.

Call DOMFD.

Exit.

3.4 DOMFD: For all files in the MDF,

If the extension is UFD,

Get the CFP (which points to the first RIB),

Read and verify the first UFD RIB.

If the RIB is acceptable,

For all files in the UFD,

Perform an extended lookup (through FILSER), print if it fails.
Get the CFP (which points to the first RIB).

Read and verify the first RIB and all extended RIBs, print if an error occurs.

If any clusters are non-existent, print and don't check the SAT's.

Checksum each group (unless RIPABC=1), print if an error occurs.

Mark the clusters allocated to this file in the computed SAT.

If any clusters are already allocated or have a bit set in the trouble SAT (for pass 2),

Set the bits in the trouble SAT, and

Print MULTIPLY-USED.

If any clusters are not allocated in the disk SAT, print FREE.

Continue.