

PROGRAMMING SPECIFICATION

DSKLST

Date: 13-Jun-73
File: DSKLST.RNO
Edition: 2

This document reflects the software of version 32A.

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 including of DEC's copyright notice) only for use in such system, except as may otherwise be provided in writing by DEC.

1.0 INTRODUCTION

This document describes DSKLST, a program that compiles and prints an analysis of the contents and the current status of the system's disk-like storage devices.

2.0 FUNCTIONS

The DSKLST output is designed to aid system administrators, operators, field service personnel, and programmers in analyzing the contents of all file structures. The program may be used to:

1. Analyze efficiency of disk space use
2. Detect and diagnose problem areas on the disk
3. Determine the need to refresh
4. Determine the optimum cluster size

3.0 OUTPUT DESCRIPTION

The DSKLST output provides a comprehensive analysis of the contents and status of the system's file structures. The following paragraphs describe each of the elements of the output listing and point out some common operating and administrative uses.

3.1 Identification Section

3.1.1 The ID Section contains the file structure name, physical unit name, unit ID name, and the logical unit name.

3.1.2 This section provides the system manager, operator, and programmer with the file structure organization.

3.2 Home Block

3.2.1 The HOME block specifies all file structure parameters, such as the location of the MFD, the number of blocks allocated for swapping (by unit), and the number of blocks per cluster.

3.2.2 The file structure's vital statistics contain much of the information specified during the once-only dialogue. Therefore, a user may request an octal dump of the HOME block for diagnostic purposes if he is in doubt as to the monitor characteristics constructed during the once-only dialogue.

3.3 File Information

3.3.1 The User File Directory (UFD) listings provide a complete directory of named files for each user, beginning with the Master File Directory (UFD 1,1). There are twelve entries for each named file.

1. Filename
2. Extension (EXT) (An asterisk following the extension means that the file is extended.)
3. Compressed file pointer (RETRVL POINTR)
4. Most recent access date
5. File creation time and date
6. Access code privileges (PRV)
7. Data mode
8. Number of words written on a file
9. Number of blocks allocated to the file
10. Error bit code (The key to the error bit code is printed just prior to the MFD listing.)
11. Tape label, if any
12. File version number, if specified.

The total allocated disk space (in blocks and in words), and the average file size (in blocks) are summarized for each user. The number of extended files is reported at the end of the summary.

3.3.2 The file information in the directory section may be used for maintenance purposes. The system administrator uses the information to redistribute space in order to make more efficient use of the disk. Individual user quotas may be modified, based on the user's total file usage. Infrequently accessed files may be deleted. The number of blocks allocated and the frequency of error bits may indicate a need to refresh.

3.4 SAT Block

3.4.1 The Storage Allocation Table reflects the current status of every available block on the disk. The SAT block is a map of the contents of a file structure, where each bit represents a cluster. A bit on indicates that the corresponding cluster is allocated, is written, is bad, or is nonexistent. A bit off indicates that the corresponding cluster is available to be written on. This section specifies the number of free blocks left in each SAT block and summarizes the total amount of disk space remaining in the file structure.

3.4.2 The SAT block is a useful maintenance tool. At a glance the system manager can determine what portions of the file structure are in use and how fragmented this usage is. He can use the number of free blocks as an indicator of when to delete files in order to create more free storage.

3.5 BAT Block

3.5.1 The Bad Allocation Table enumerates bad regions on a particular unit and specifies the origin and length of these bad regions. The BAT block indicates the total number of bad sectors, bad regions (as determined by both the MAP program and the monitor), and the controller device code for each unit. The extended BAT listing specifies, for each bad

segment, the physical hardware location of the bad region. This includes the location of the cylinder, the surface, and the sector or block in both octal and decimal. The extended BAT block also specifies the physical unit name, the processor and the status of the controller at the time the trouble was discovered. An asterisk indicates that this region was previously determined bad by another controller. Output of this nature is especially useful to field service personnel for diagnostic purposes.

3.5.2 The BAT block shows the system manager or operator how well his disks are performing and the extent of trouble if performance is poor. It is an instrument to help diagnose problems on a disk pack. A pattern may emerge that is indicative of the nature of the trouble. The BAT block can also be used as an indication of the need to refresh.

3.6 Error Summary

3.6.1 DSKLST prints a statistical summary of file structure errors including:

1. The number of words wasted in partially written blocks, expressed as a percent of the total words written.
2. The number of blocks wasted in partially written clusters, expressed as a percent of the blocks written. (This does not include blocks which were allocated, but left completely unwritten.)
3. Discrepancies in the number of free blocks as:
 - a. Computed by subtracting the number of allocated blocks in the file structure from the total number of blocks in each unit.
 - b. Returned by the DSKCHR UOU.
 - c. Counted by the SAT blocks. DSKLST counts the number of free bits and multiplies this by the cluster size. The result of this computation is the governing number. The system continues to allocate free storage until the SAT is filled.

These three methods of computing the number of free blocks produce different results due to inconsistencies in the methods of computation. Legitimate differences can occur after crashes, since certain blocks may be lost as a result of SAT updating prior to directory updating.

4. A summary of the number of files with hard errors in each of the following categories:
 - a. Hard write - Either a device or data hardware write error.
 - b. Hard read - Either a device or data hardware read error.
 - c. Soft check - A software checksum or redundancy error.
 - d. Damage assess - Not used. This entry is reserved for tallying errors found by a damage assessment program.
 - e. Failsa - Not used. This entry is reserved for tallying errors found by the FAILSAFE program.
 - f. Crash - Not used. This entry is reserved for tallying the number of partially written files closed after the monitor stops.

3.6.2 These error statistics provide the user with a useful summary of disk characteristics and problems. The first two entries (percentage of wasted words and blocks) are an indication of the cost of clustering and is one of the guides used to determine optimum cluster size. (Refer to paragraph 4.0). The number of file blocks remaining is an indication of the need to refresh.

3.7 Performance Summary

3.7.1 The logical histogram is a frequency distribution table that lists the total number of files for each file size (from 0 to .GE. 51 blocks in length). The file size is given in "blocks written" and does not include any RIB blocks.

3.7.2 This table is very useful in determining optimum cluster size. (See paragraph 4.0 below).

4.0 DETERMINE OPTIMUM CLUSTER SIZE

File storage space is assigned and released in clusters of blocks rather than in single blocks. The cluster size is determined by the system manager and established by the operator at once only. For maximum efficiency it is recommended that each unit have all its SAT blocks in core. Clustering reduces the amount of core required to store the SAT blocks. The larger the cluster size, the fewer the number of SAT blocks required for storage allocation information, the fewer the number of SAT reqd/writes, and the fewer the operations required to assign and release space. Although large clusters save core, it is usually at the expense of disk space. Since an entire cluster is automatically allocated to each named file, short files will result in wasted blocks if the cluster size is large.

To determine the optimum cluster size, the manager refers to several entries on the DSKLST output. The first is the logical histogram, since this table summarizes the lengths of all files within a file structure. The error summary indicates the number of wasted blocks, based on the number of blocks allocated but left partially unwritten. The file directory gives the average number of blocks per file for each user.

For example, if most of the files are 2 or 3 blocks in length, the statistics suggest that a cluster size of 5 is appropriate. (Two or three blocks in length plus the 2 overhead blocks = 5). The manager then computes the cluster size based on the amount of core required to store and manipulate the SAT blocks. In our example with a cluster size of 5, a file structure of 40000 blocks can be represented by 2 SAT blocks.

$40000 \text{ blocks} / 5 \text{ blocks per cluster} = 8000 \text{ clusters}$

Each SAT block has 128 words - 10 words reserved for system information (5.03 monitor only) = 118 words

$118 \text{ words} * 36 \text{ bits per word} = 4248 \text{ bits}$

Therefore, 2 (4248 bit) SAT blocks are needed to represent 8000 clusters.

Storing and manipulating two SAT blocks in core is acceptable. Therefore, 5 is the optimum cluster size for this file structure.

The optimum cluster size is a balance between the amount of core required (large clusters) and the amount of disk space wasted (small clusters). A cluster size of 10 for a file structure with any RP02's yields a SAT.SYS file in which each unit has only one SAT block. A cluster size of 1 for structures with RM10B's and RD10's yield the same result. It may be advisable to increase the number of SAT blocks per unit for large disks, like the RA10, or in cases where wasted disk space is unacceptable.

5.0 OPERATING INSTRUCTIONS

5.1 Assembling DSKLST

The operating instructions for DSKLST are very simple. It is a one (low) segment program that has no assembly options. The user assembles the source program with MACRO, loads, and saves it with the following dialogue. (Note that TTY output is underlined in order to distinguish it from user input.)

```
.COMPILE DSKLST
```

```
-
```

```
MACRO: DSKLST
```

```
-----
```

```
.LOAD DSKLST
```

```
-
```

```
MACRO: DSKLST
```

```
-----
```

```
.SAV dev:DSKLST
```

```
-
```

```
.JOB SAVED
```

```
-
```

5.2 Running DSKLST

5.2.1 Under ordinary circumstances to run DSKLST with the output printed by a line printer. the user need not preassign a device, DSKLST writes the output on logical device LPT.

5.2.2 The user initiates the execution of DSKLST by typing

```
.R DSKLST
```

```
-
```

The program returns

```
/H FOR HELP
```

```
-----
```

```
*
```

```
-
```

asking the user to type in the file structure names separated by commas, a colon and one or more switches to select the appropriate segments of the listing. Finally, a carriage return terminates the command string. If no file structures are specified, all file structures are listed. If no switches are specified, the entire DSKLST output is printed.

5.2.3 If the user prefers to print his output on another device, he must prespecify the device through logical device assignment. For example, to print portions of the DSKLST output on a terminal, the user types

```
.ASSIGN TTY LPT
```

```
-
```

to which the system returns

```
TTYn ASSIGNED
```

```
-----
```

indicating that the terminal has been successfully assigned and the user is free to begin execution. (See paragraph 5.2.2.)

5.2.4 For complete, lengthy output, it is recommended that the output is written on the disk and later spooled to the line printer. This technique, available with the 5.03 monitor, increases line printer speed and efficiency. Earlier monitors must pre-assign the disk in the same way the Teletype was assigned in paragraph 5.2.3, above.

Before initiating the execution of DSKLST the user types

```
.SET SPOOL LPT
```

```
-
```

Next the user initiates execution of the program as in paragraph 5.2.2. When the program has finished compiling and writing the DSKLST.LST file on the disk, the system returns

EXIT

.

Normally the listing is queued and later spooled to the line printer in its turn. For an immediate listing the user types

.QUEUE *

-

5.3 Switches

Each of the switches selects and prints a subset of the DSKLST output for each file structure. These switches are described below.

- /U Prints the identification section including the file structure name, physical unit name, unit ID, and the logical unit name.
- /V Prints the identification section and the HOME block, an octal dump of the file structure's vital statistics.
- /F Prints the File Information Section beginning with the Error Bit Key, through the entire list of User File Directories.
- /S Prints an octal dump of the SAT blocks.
- /B Prints an octal dump of the BAT blocks and a list of all the error regions.
- /E Prints the summary of error statistics.
- /P Prints the summary of performance statistics.
- /Q Prints quick listing, using normal LOOKUPS instead of extended LOOKUPS.
- /N Prints the DSKLST output on narrow paper, eliminating the RETRVL POINTR, TAPE LABEL, and VERSION columns of the UFD listing. TTY output automatically assumes the narrow version.

/H Types a TTY explanation of these switches for the user.

5.4 Recommended Usage

It is recommended that the system operator run the entire DSKLST output under project-programmer number 1,2 at least once per week, but preferably more often.

The BAT block (/B) should be printed daily and kept in a binder for reference. Most of the DSKLST output can be efficiently compiled and listed in a timesharing environment, under any project-programmer number. The SAT block is an exception. It should be listed stand-alone since the status bits change as files are created and destroyed in a timesharing environment. Also note that project-programmer [1,2] has universal access privileges. Attempts to read protected files under another project-programmer number result in a LOOKUP FAILURE error message.

5.5 Associated Terminal Messages

CAN'T OPEN OUTPUT DEVICE

The device is not available. e.g., The LPT is in use by another job.

CAN'T ENTER OUTPUT FILE

The Enter failed. e.g., The disk is full.

NO SUCH FILE STRUCTURE

The file structure does not exist.

