FastTrax ™

The Best Disk Optimizer

Version 4.0

- Includes MakeTrax™
- Speeds Up Hard Disk Performance
- Extends Hard Disk Life
- Allows Flexible File Placement

FastTrax International

FastTrax™
User's Guide

Contents

Introducing FastTrax

Overview

FastTrax and its companion program MakeTrax are easy-to-use programs that can significantly reduce the time required for accessing your hard disk or floppy disks.

The problem is that DOS is not a very neat housekeeper; it makes very little effort to keep files physically in one contiguous piece. These fragmented files result in slower access. The purpose of optimization is to collect these non-contiguous pieces into one contiguous group of clusters for each file. (The terms *fragmented file*, *contiguous file*, and *cluster* are explained in *Definitions* in *Appendix B*.)

For example, with a cluster size of 2K (2,048 bytes), a fragmented 64K file could reside in as many as 32 separate clusters spread over the disk. Although this isn't very likely, a file of that size will often be in four or five non-contiguous cluster groups. To reference this file, the operating system must reposition the read/write head for each cluster group, and this positioning process is one of the most time consuming, non-productive disk activities.

FastTrax optimizes hard disk and floppy disk performance by consolidating scattered pieces of files and directories. FastTrax not only allocates files to contiguous disk clusters, it assigns most files to the least number of disk cylinders, thus reducing track-to-track seeks.

In addition to defragmenting your disk, you can direct FastTrax to place your frequently used files on the "fast tracks" of your disk -- near the File Allocation Table (FAT) -- and place your infrequently used files at the opposite end of your disk.

The MakeTrax program helps you to quickly specify the placement of every file and directory on your disk for optimum performance. MakeTrax also displays fragmentation data for each file and directory.

FastTrax is compatible with existing copy-protection schemes. FastTrax itself is not copy-protected.

Limits

If you have enough memory, FastTrax will optimize any disk that DOS supports, from floppy disks to hard disk partitions larger than 512MB. There are no limits on the number of files, directories, directory levels, or number of clusters, except those imposed by DOS. See *Requirements* and *Determining If You Need Expanded Memory*.

The FASTINFO.DAT File

This FASTINFO.DAT file on your FastTrax disk contains last minute information that may not be in your manual. It also contains a list of files on your FastTrax disk. You can read this file by typing FASTINFO. Please read this file before running FastTrax.

The FASTINFO program also checks the integrity of your FastTrax program files.

What's New

In response to comments from our users, FastTrax and MakeTrax have several new features:

o   Increased capacity. By using expanded memory (EMS), FastTrax can optimize disks with any number of files, directories, and clusters. FastTrax automatically utilizes your computer's conventional and EMS memory. Prior FastTrax versions (below 4.00) are limited by the amount of conventional memory in your computer.

o   MakeTrax command file editing. MakeTrax now allows you to modify an existing command file.

o Memory testing. As an extra precaution, FastTrax and MakeTrax now check for faulty memory locations in both conventional and expanded memory (when needed) before starting any function.

o New handling of read-only files and directories. By default, FastTrax does not move *read-only* files and directories. But now you can override this default with a new FASTTRAX.CM0 command. Similarly, MakeTrax now allows you to specify the placement of *read-only* files and directories. FastTrax still does not move files and directories with the *hidden* or *system* attribute.

o New *clear* feature. This option clears (with zeros) all free clusters and removes all unused directory entries, preventing access to deleted confidential data.

o New *optimize directories only* feature. This option quickly removes unused directory entries.

o Improved determination of drive parameters. FastTrax and MakeTrax now use an improved algorithm to determine the physical parameters of your drive (such as number of heads, and sectors per track).

o Improved speed. Both FastTrax and MakeTrax are now faster.

Requirements

o An IBM PC, XT, AT, PS/2, or compatible. See *Appendix C - Configuring FastTrax*, for more details.

o PC-DOS or MS-DOS, version 2.0 or higher.

o Memory requirements:

256K RAM for floppy disk systems.

512K RAM for most hard disk systems.

640K RAM plus 800K expanded memory (EMS) may be needed for hard disk partitions larger than 85MB.

Determining If You Need Expanded Memory

If you computer system has at least 800K of expanded memory, you can skip this section because you have enough memory to optimize any disk.

Since FastTrax's memory requirements are not directly related to drive partition size, there is no simple way to know if FastTrax will need expanded memory to optimize your disks. However, since FastTrax always ensures there is enough memory before proceeding, the easiest way to know if FastTrax will optimize a disk is to just try it. If that's not practical, here are some guidelines to you estimate memory usage:

o   If you own a previous version of FastTrax, and it is able to optimize all your drives, you probably don't need expanded memory.

o   If your largest disk partition is smaller than 85MB, you probably don't need expanded memory.

o   If your largest disk partition is larger than 85MB, you might need expanded memory.

o   If your largest disk partition is larger than 500MB, you probably do need expanded memory.

FastTrax and Copy-Protection Schemes

To maintain compatibility with copy-protection schemes, FastTrax never moves a file or directory that has *hidden* or *system* attributes. FastTrax also does not move *read-only* entries unless specifically instructed to do so. Even so, since some software companies regularly change their protection schemes, you may want to "un-install" copy-protected software before running FastTrax.

In theory, there may be instances in which copy protection is based upon the location of certain files other than those with *hidden* or *system* attributes. In this case, FastTrax may move these files and render the copy-protected program unusable, unless these files are specifically "held" in the FASTTRAX.CM0 file. (See *Getting the Most Out of FastTrax* for a description of how to "hold" such files.)

The safest approach is to check with the publisher of the copy-protected program before using FastTrax.

Disk Terminology

The words *disk* and *drive* are used interchangeably. The terms *sectors*, *tracks*, *clusters*, and *cylinders* are all units of disk space. See *Definitions* in *Appendix B* for basic definitions and some graphic illustrations. You don't need to understand every term to use FastTrax effectively.

Safety

We've done our best to make FastTrax as safe as possible. However, you may lose valuable data if your computer system is not operating properly. Some disk and memory problems may show up only when running FastTrax because FastTrax often uses all available memory and usually references every area of your disk. If you have any hardware or software problems, they will probably show up the first time you use FastTrax.

Before running FastTrax (or any other disk optimizer) the first time, we urge you to:

o   Back up your disk!

o   Read *Appendix D - Recovery* to familiarize yourself with the recovery procedure.

FastTrax is designed to protect your data even if your computer system loses electrical power during the optimization process. In most cases, you will not lose any data. See *Appendix D - Recovery* for more details.

After a typical failure, the DOS CHKDSK function may display many "Invalid directory entry" messages; it may also report lost clusters. However, your files are still intact and you can recover all your data.

Resident Programs

Before running FastTrax, you must remove or disable all memory resident programs, background programs, "terminate-and-stay-resident" programs" and multi-tasking programs that may access the disk being optimized. You don't need to remove the DOS VDISK.SYS and ANSI.SYS device statements from your CONFIG.SYS file.

The simplest way to avoid resident programs is to run FastTrax from a floppy disk. See FastTrax Basics, *Installing FastTrax*, for specific instructions.

You can run FastTrax from your hard drive as long as your resident programs do not interfere with FastTrax while it is working. If you have a good understanding of your resident programs, you may want to experiment to see if FastTrax will run safely while they are in memory. But back up your drive first!

If you're not sure you have any resident programs, use the DOS TYPE command to list your CONFIG.SYS and AUTOEXEC.BAT files, since resident programs are often loaded from these two files as well as from the DOS command line. Resident programs include "pop up" programs, device drivers, local area networks, and "multi-tasking" environments.

Since some resident programs contain errors that corrupt memory used by FastTrax, the safest policy is to avoid all non-essential resident programs.

You should particularly avoid programs that change the memory refresh rate of your computer. Examples of such programs are QFRESH.COM (described in the July 1988 issue of PC Magazine), NREFRESH.COM, SPEEDER.COM and MHZ.COM. These and others can be downloaded from many computer bulletin boards.

These programs can enhance computing performance slightly by decreasing your computer's memory refresh rate. However, if the refresh rate is too low, your computer system will become unstable. Some programs may run properly, but others may not. Since there is no easy way to create a "worst case" memory timing test, rigorous testing is needed to find a "safe" rate. Bear in mind that a safe rate may become unsafe if you change hardware options, or as your system ages.

Local Area Networks

You can download FastTrax from a network server disk to optimize a local disk that is not on the network.

Do not use FastTrax to optimize a network disk when the network is operational. While some networks prevent access by utilities like FastTrax, there are some networks that allow direct access to network disks.

To optimize a network disk, boot DOS on the computer with the network disk, but do not start the network software. Run FastTrax, then start the network software.

ASSIGNed, SUBSTed, and JOINed drives

If you use the DOS ASSIGN command to reassign a drive, FastTrax follows the indicated assignment. For example, if you assign drive C to drive D and instruct FastTrax to optimize drive C, FastTrax will optimize drive D.

If you use the DOS SUBST command to substitute a drive letter for another drive or path, FastTrax will optimize the entire drive associated with the assignment. For example, if you substitute drive letter G for C:\reports\files and instruct FastTrax to optimize drive G, FastTrax will optimize all of drive C, not just the files in C:\reports\files.

If you use the DOS JOIN command to join a drive to a directory on another drive, FastTrax will allow you to optimize the joined drive (even though DOS no longer recognizes it as a drive).

We recommend that you optimize "real" drives only.

Technical Support

Technical support is available Monday through Friday from 9:00 a.m. to 5:00 p.m., Pacific Time. Please call or write:

FastTrax International
8 Digital Drive, Suite 102
Novato, CA 94949
(415) 883-9561

FastTrax Basics

Installing FastTrax

To install FastTrax on a floppy disk:

Since the FastTrax files require about 360K, you'll need either a high-density 5 1/4" floppy disk or a 720K or larger 3 1/2" micro disk.

1. Format an unused floppy disk with the /S switch.

2. Copy the FastTrax files to the formatted floppy disk.

3. Store your master FastTrax disk in a safe place.

4. Create CONFIG.SYS with BUFFERS=6. (You can use a different number of buffers or omit the BUFFERS command entirely; FastTrax doesn't need buffers for good performance. Each buffer uses about 530 bytes of RAM. See your DOS manual for more information about the CONFIG.SYS file.)

5. If your hard disk requires a special disk driver, include the required DEVICE= command in your CONFIG.SYS file.

6. Finally, boot from this floppy disk.

To install FastTrax on a hard disk:

1. Place your FastTrax disk in drive A.

2. To install FastTrax on hard drive C, type *A:INSTALL C:* and press *Enter*. To install FastTrax on a different hard drive, type its drive letter instead of the letter C.

3. Remove your FastTrax disk from drive A and store it safely.

Quick Start

If you are familiar with DOS and don't want to read the entire manual, proceed as follows:

1. Back up your disk before running FastTrax the first time! (See *Safety* under *Introducing FastTrax*.)

2. Remove or disable any memory resident programs, including local area networks, that might use the disk that FastTrax will optimize.

3. If you want to assign your files and directories to specific areas of your disk, type *MAKETRAX* at the DOS command line and follow the directions displayed by MakeTrax.

4. To run FastTrax, type *FASTTRAX* at the DOS command line and follow the directions displayed by FastTrax.

Running FastTrax

The following paragraphs show you how to run FastTrax. For more technical information, read *Appendix B - About Disk Optimizing*.

To run FastTrax, type *FASTTRAX* and press *Enter*.

FastTrax prompts you for the "target" drive to analyze. To bypass this prompt, include the target drive letter and a colon on the command line (e.g., *FASTTRAX C:*).

FastTrax then analyzes the target drive and displays a visual fragmentation report.

FastTrax now asks you "Do you want to continue (Y/N)?". If you continue, FastTrax displays a list of functions:

A   optimize All files and directories on your disk.

Use this function the first time you optimize a disk or when the disk is badly fragmented. See *Appendix B - About Disk Optimizing*, for technical details.

F   optimize Fragmented files and directories only.

Use this function to touch up your disk. It's much faster, but it may leave a few fragmented files if your disk is nearly full or if the fragmented files are very long. This function looks for a better place for your fragmented files. Since it doesn't pack your disk, it may leave a few free space holes.

D   optimize Directories only.

Use this function to quickly remove deleted directory entries from all directories.

C   Clear all deleted data.

Use this security function to clear (with zeros) all remnants of deleted file data, and remove all deleted directory entries from your disk.

X   eXit without doing anything.

If you choose the A or F function, FastTrax looks for a file named FASTTRAX.CM0 in the root directory of the target drive. The section *Getting the Most Out of FastTrax* describes the FASTTRAX.CM0 file and how to create it. If found, FastTrax displays the first few lines of this file and asks you if you want to use it.

If you answer Yes, FastTrax uses the information in this file for the directory and file placement order during the optimization. If you answer No, or if the FASTTRAX.CM0 file is not found, then FastTrax uses the default optimization order. The default order is described in *Appendix B* under *Packing Methods*.

Before starting the function you requested, FastTrax asks you to choose a Data Checking option. Your choices are:

No checking	This option is safe and fast if your computer system is reliable.
Verify	This option uses the DOS VERIFY feature to confirm that all data written to disk can be read. The data itself is not checked. We recommend this option because it offers a good combination of speed and data security.

Compare   This option directs FastTrax to read and check all data it writes to your disk. Though slower, it offers the highest level of data security.

FastTrax then performs the requested function and displays status messages showing what it is doing. You can interrupt FastTrax at any time by pressing the *Esc* key.

When FastTrax is finished, it asks you if you want to see the new organization. If you answer No, FastTrax displays the opening menu so you can select another drive (or exits to DOS if you specified a drive letter on the command line).

Rebooting After Optimizing

Before returning you to DOS, FastTrax prompts you to reboot your computer before using a drive that you just optimized. Rebooting will reinitialize resident programs that still expect your files and directories to be where they were before optimization.

One such resident program is DOS's FASTOPEN program. FastTrax specifically looks for FASTOPEN and warns you if it is resident, in which case you <u>must</u> reboot if FASTOPEN controls <u>any</u> of the drives you have just optimized.

If you're not using this type of program, you don't need to reboot. If you're not sure, then be safe and reboot.

You can suppress reboot prompts using the */NR* command line option (described next).

Command Line Options

You can run FastTrax from the DOS command line or from a batch file using the command line options described in this section. In addition to the description below, FastTrax outputs a brief list of its options when you type *FASTTRAX ?* at the DOS command line.

Items in *[..]* are optional; do not type the square brackets. To run FastTrax with command line options, type *FASTTRAX* followed by one or more options:

FASTTRAX [d:] [/R] [/NR] [/Ex:] [/O] [/P]
[/B[YNAFDC123P]]

where:

d: Drive letter.

        The default is the currently logged drive.

/R   Report only.

        FastTrax displays the fragmentation report and then returns to DOS.

/NR   No reboot prompts.

        For batch operation, this option suppresses automatic rebooting after optimizing.

/Ex: Specifies the drive letter used by an EMS emulator.

        If you are using an EMS driver that emulates expanded memory on a hard disk (such as Above Disc from Above Software, Inc.), use this option to specify the drive letter *x:* of the emulated disk.

        FastTrax attempts to detect EMS disk emulation and -- unless you specify its disk location -- does not use EMS, to avoid possible data loss during optimization (i.e., optimizing a disk while the EMS driver is also using it). By specifying the drive letter, FastTrax uses EMS provided there is no conflict.

        You can also specify the emulated drive letter with the DOS SET statement:

        SET FT_EMSDRV=x: (with no extra spaces)

        By placing the SET statement in your AUTOEXEC.BAT file, you don't need to specify the */E* option each time you run FastTrax.

        If your expanded memory is not disk-based, you don't need the */E* option or the SET statement.

/O   Output file list.

This option directs FastTrax to output a list of all the files and directories on the specified drive. You can use this list as the basis for building a custom FASTTRAX.CM0 file.

However, unless you have very special placement requirements, you should use the MakeTrax utility to build FASTTRAX.CM0. See *The MakeTrax Utility* in *Getting the Most Out of FastTrax*.

Note: Using the DOS redirection operator >, you can save the list in a disk file. For example, to save the complete file and directory list of drive D in D:\FASTTRAX.CM0, type:

FASTTRAX D: /O >D:\FASTTRAX.CM0

/P   Output disk parameters.

See *Appendix C - Configuring FastTrax*.

/B   Batch operation.

The batch options correspond to the responses to some of the prompt messages that FastTrax outputs during interactive operation. The batch options are:

Y   Use the FASTTRAX.CM0 file if found (default).

N   Do not use the FASTTRAX.CM0 file.

A   Optimize ALL files on the disk (default).

F   Optimize the FRAGMENTED files only.

D   Optimize the DIRECTORIES only.

C   CLEAR all deleted directory entries and deleted files.

1   No verify.

2   DOS verify (default).

3   Compare data.

P   Pause if FastTrax detects any error.

It's a good idea to use the *P* option when running FastTrax from a batch file so you can read the FastTrax error messages before the batch file continues.

Running FastTrax from a Batch File

The /B option described above allows you to run FastTrax without any prompts. It also allows you to optimize one or more disks from a batch file by including a *FASTTRAX* command for each disk.

When running FastTrax with the /B option, FastTrax reboots your computer system immediately after completing the requested optimization unless you include the /NR option on the command line. Therefore include the /NR option (no reboot) on all but the last *FASTTRAX* command in the batch file. For example, to optimize drive D, clear drive D with no verify, and optimize drive E, your batch file might contain:

    FASTTRAX D: /B   /NR
    FASTTRAX D: /BC1 /NR
    FASTTRAX E: /B

FastTrax Return Codes

When running FastTrax from a batch file, you can test the return code with the DOS *IF ERRORLEVEL* batch file command. The return codes are:

| Code	Meaning
0	Successful completion.
1	Disk read error.
2	Disk write error.
3	Inconsistent data (similar to the conflicts that the DOS CHKDSK command detects).
4	Not enough RAM.
5	Syntax error.
6	Miscellaneous error.
7	Internal error.

The Fragmentation Report

This visual report shows the current state of the disk. Each map symbol represents a small group of clusters. If all the clusters in a group have a common characteristic (e.g., *free*), the report displays that characteristic. If the clusters have more than one characteristic, FastTrax displays the highest priority characteristic according to the following table:

| Priority	At least one cluster ...
highest	is bad
	belongs to a held file
	belongs to a fragmented file
	belongs to a consecutive file
	belongs to an optimized file
lowest	is free

For example, if a cluster group contains three free clusters, four consecutive clusters, and one bad cluster, FastTrax displays that cluster group as *bad*.

Since fragmentation often occurs near bad or held clusters, the fragmentation map usually reveals the cause of the fragmentation.

The following illustrations show fragmentation for a typical hard disk before and after optimizing:
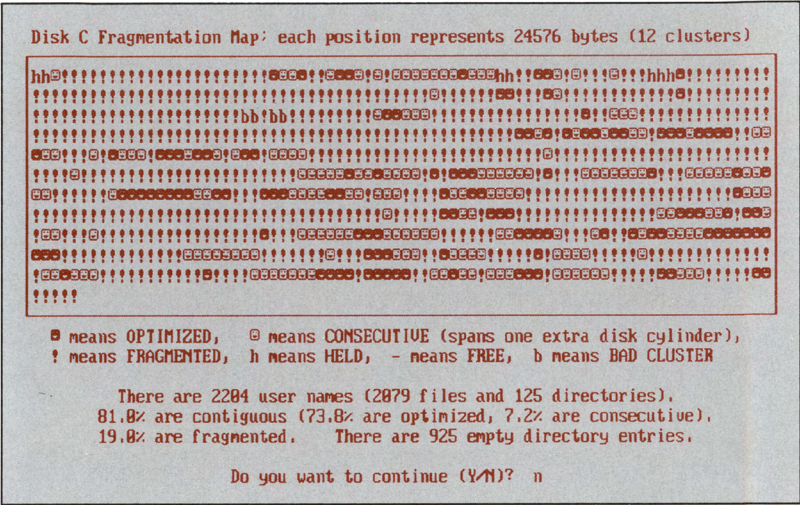
FRAGMENTATION REPORT BEFORE OPTIMIZING
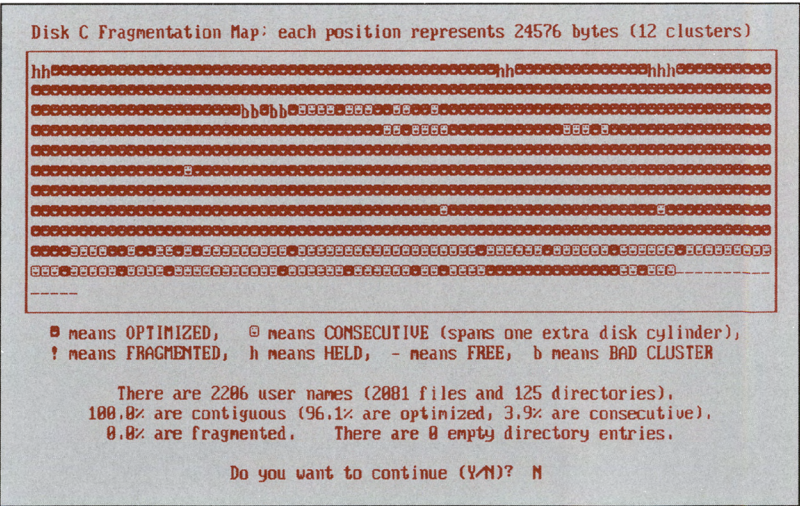
FRAGMENTATION REPORT AFTER OPTIMIZING

Getting the Most Out of FastTrax

Your FastTrax disk optimizer will further improve disk performance if you create a FASTTRAX.CM0 file in the root directory of each disk to specify the best placement for your files and directories. The MakeTrax utility helps you create the FASTTRAX.CM0 file.

The MakeTrax Utility

The MakeTrax utility analyzes your disk and displays a list of all your files and directories, showing the fragmentation data and current placement for each item. You can then scroll through the list and assign new placement priorities for individual items or groups.
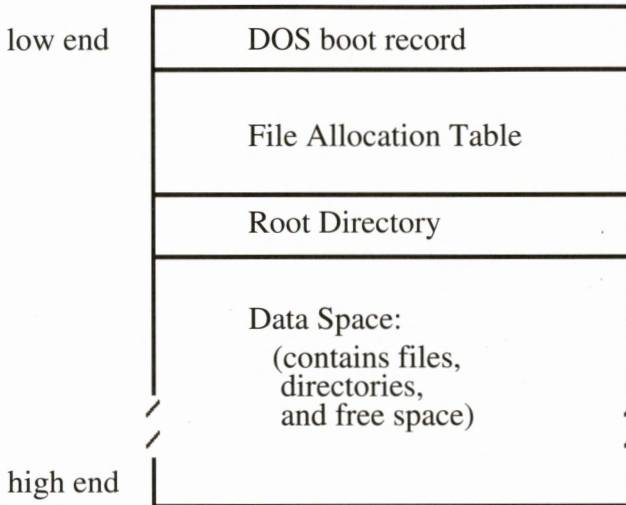
After you assign your priorities, MakeTrax shows you the size of each priority group and displays five placement strategies. Based on your needs, you can select one of the strategies.

MakeTrax then builds FASTTRAX.CM0 in the root directory of the selected drive. You can repeat the operation for other disks, run FastTrax or exit to DOS. When you run FastTrax, you can direct FastTrax to use the FASTTRAX.CM0 file when you select an optimizing function.

Organizing Your Disk

To use MakeTrax effectively, it helps to understand the general layout of your disk.

DOS maintains your directories and files within the bounds of a DOS *partition* on your disk. DOS allows more than one partition on a hard disk. The format of a DOS partition is shown below, starting from the low end of the partition:

low end	DOS boot record
	File Allocation Table
	Root Directory
high end	Data Space: (contains files, directories, and free space)

The File Allocation Table (FAT) helps describe the placement of each entity in the Data Space. When accessing a file, DOS must reference both the FAT and the file itself. If the file and the FAT are physically located far apart from each other, the read/write heads in your disk drive must travel long distances when shuttling back and forth between these structures. This translates into long access times and poor performance.

Since the FAT resides at the low end of the disk, access times to files at the low end of the disk are much faster than to files at the high end of the disk. Therefore you can improve disk performance by assigning your frequently used files near the FAT and assigning your infrequently used files far from the FAT (at the high end of the disk partition). Such assignments improve the average access time to your frequently used files.

Running MakeTrax

To run MakeTrax, type *MAKETRAX* and press *Enter*. If you have difficulty reading your display, press *Esc* to exit MakeTrax, then type

MAKETRAX /L and press *Enter*; MakeTrax will use a different combination of highlights and shading that may be more readable.

If you are familiar with FastTrax and DOS and don't want to read this chapter, you can invoke MakeTrax and follow the directions displayed by MakeTrax.

MakeTrax prompts you for a drive letter. Enter a drive letter and press function key *F10*. MakeTrax then analyzes the drive.

If you previously created a FASTTRAX.CM0 file, MakeTrax processes the file so you don't have to re-enter your assignments. MakeTrax ignores references to deleted files and directories, and omits them when it creates the new FASTTRAX.CM0 file.

Note: Previous FastTrax versions used *FASTTRAX.CMD* for the command file name. FastTrax now uses *FASTTRAX.CM0* to avoid a conflict with OS/2 command file names. MakeTrax and FastTrax still recognize and process the prior name. This change accommodates users who run DOS and OS/2 on the same disk. You should not run FastTrax under OS/2.

The MakeTrax Assignment Screen

After analyzing the specified drive, MakeTrax displays a list of all your files and directories. The list starts with the files in the root directory and continues with the files in all the other directories. Each list item describes a file or directory name. Read-only files and directories are shown with an 'R' at the right of the screen.

To move around the list, use the *Up* and *Down* arrow keys to scroll one line, the + and - keys to scroll one directory, the *PgUp* and *PgDn* keys to scroll one screen, and the *Home* and *End* keys to scroll to the beginning and end of the list. Make sure that the *Num Lock* key is off.

Above the list is a disk map window which shows the current placement of the selected item in relation to the FAT and its parent directory (indicated by a D). The map also shows if the item is fragmented, consecutive, or optimized.

Assigning Placement Priorities

Use the left and right arrow keys to assign a placement priority. The priorities are:

Priority	Meaning
top	Closest to the File Allocation Table.
high	(between "top" and "normal".)
normal	(the default assignment.)
low	Farthest from the File Allocation Table.
(hold)	FastTrax will not move this item.

Don't confuse "low end of the disk partition" with "low placement priority." The FAT is at the low end of the partition, where *low* refers to the cluster numbers near the FAT. To place a file in the low clusters for fast access, assign it to the *top* placement priority.

Similarly, the high end of the partition is far away from the FAT, *high* meaning high cluster numbers. To place an infrequently used file far away from the FAT, assign it to the *low* placement priority.

The guidelines for assigning placement priorities are simple:

o   Assign frequently used items to the *top* and *high* priority groups.

o   Assign infrequently used items to the *low* priority group.

o   Assign all other items to the *normal* priority group.

Beyond these guidelines, it's usually a good idea to place each file near its directory.

Placing files near their directories is FastTrax's default ordering method. Some disk optimizers place all directories near the low end of the disk. Although we think the FastTrax method is better, if you prefer the "directories first" approach, simply assign all the directory names to the top priority (by using the *D* group command described later under *Group Assignments*.) For technical details, see *The Default Packing Order* in *Appendix B*.

Another good idea is to assign the same priority (e.g., *top*) to all directories on your DOS PATH, significantly reducing the time for DOS to search the path.

The *(hold)* Priority

You can direct FastTrax to not move a file or directory by assigning it to the *(hold)* priority.

To maintain compatibility with copy-protection schemes, FastTrax never moves files or directories with a *hidden* or *system* attribute. MakeTrax and FastTrax automatically designate items these attributes as *held*.

As an extra precaution, you can hold all the "visible" files that belong to a copy-protected program as well, on the possibility that its protection scheme expects even these files to be in their original places.

Assigning *read-only* Files and Directories

By default, FastTrax does not move files and directories with the *read-only* attribute. Also by default, MakeTrax assigns the *(hold)* placement priority to all read-only items. By pressing function key *F8* (*move R/O*), MakeTrax allows you to change their priority. By pressing *F8* again, MakeTrax reassigns the *(hold)* placement priority to all read-only items.

The *(held)* Priority

Since FastTrax allows you to move entries in a directory marked as *hidden* or *system*, MakeTrax displays such directory names with the special *(held)* priority. The *(held)* directory priority is fixed, but you can change the placement priorities of all non-held files and directories within the *(held)* directory.

Group Assignments

After assigning a placement priority, you can assign the same priority to other files and directories by pressing a group key:

Key	Meaning
A	assign All subsequent files in the same directory
E	assign all subsequent files with the same Extension
F	assign all subsequent Files
N	assign the Next item
S	assign all subsequent files in the same directory and all its Subdirectories and their files
D	assign all subsequent Directories

The *E* and *F* assignments apply when the selected item is a file. The *S* and *D* assignments apply when the selected item is a directory.

To assign all the items in a group (starting from the beginning of the list rather than just the subsequent items), press function key *F5* before pressing *A*, *E*, *F* or *D*.

MakeTrax usually performs group assignments very quickly. But it will complete a group assignment even more quickly if you press the space bar to suppress screen updates during the assignment.

Examples of Group Priority Assignments

To place all the files in the current directory very close to the FAT, assign the first file to *top* priority and press *A*.

To place all the *.DAT files farthest from the FAT, assign any *.DAT file to *low* priority, press *F5*, then press *E*.

To assign all remaining *.EXE files to *high* priority, assign the current *.EXE file to *high* priority and press *E*.

To assign the next file or directory to the same priority as the selected file or directory, press *N*.

To place all files in the selected directory and all its sub-directories and their files farthest from the FAT, assign the selected directory to *low* priority and press *S*.

To assign all remaining directories to *high* priority, assign the current directory to *high* and press *D*.

To place all the directories near the FAT, assign any directory to *top* priority, press *F5*, then press *D*.

The Undo command

If you make a mistake, press *U* to undo the last operation.

Assignment Screen Command Key Summary

Key	Meaning
Up/Down	up/down one item.
PgUp/PgDn	up/down one screen.
+/-	next directory / prior directory.
Home/End	first/last item.
Left/Right	change placement priority.
A	assign All subsequent files in the same directory.
E	assign all subsequent files with the same three letter Extension name.
F	assign all subsequent Files.
N	assign the Next item.
S	assign all subsequent files in the same directory and all its Subdirectories and their files.
D	assign all subsequent Directories.
F5	next *A*, *E*, *F*, or *D* key assigns all items (not just subsequent items); press *F5* again to cancel.
F8	toggle move/hold read-only files.
U	Undo last operation.
F10	select placement strategy (after all assignments).

Selecting a Placement Strategy

After assigning all priorities, press F10 to select a placement strategy. MakeTrax calculates the size of each priority group and displays five placement strategies.

The initially highlighted strategy is a suggested placement strategy. If MakeTrax found an existing FASTTRAX.CM0 file, MakeTrax highlights the strategy selected in the file.

The priority group table shows the size of each group: *free* is the total available free space on the disk; *bad* is the size of the unusable disk space; *held* is the size of the directories and files marked as *hidden* or *system*. FastTrax will not move held items.

Use the left and right arrow keys to select a placement strategy. You can also select a strategy by typing its number.

The best placement of your priority groups in relation to the free space on your drive depends on your needs. MakeTrax offers five strategies:

Strategy	When to use it
1	Use when your disk is nearly full. It offers fast access to your frequently used files and to the free space.
2	Use if your programs are rapidly creating new files. (DOS places new files and directories in the free area.)
3	Use if your drive is not very full, and will probably stay that way. This strategy assumes that the free space at the high end of the disk will not be used before FastTrax is run again.
4	This compromise strategy offers fast access to your frequently used files and the first free space, and good access to your *normal* priority files. But two free spaces may encourage later fragmentation. See *Assigning Free Space* in *Appendix B* for more details.
5	Use if your disk is nearly full. Similar to Strategy 1, but offers faster access to your *normal* priority files.

Creating the **FASTTRAX.CM0** File

After you have selected a placement strategy, press function key *F10* to direct MakeTrax to build, display, and save your placement priorities and selected strategy in \FASTTRAX.CM0 on the target drive. Your original command file is renamed FASTTRAX.BAK.

After building FASTTRAX.CM0, MakeTrax returns to its opening screen. Now you can run FastTrax by pressing function key *F9*, or select another drive, or return to DOS.

Creating a Special FASTTRAX.CM0 File

While MakeTrax is able to create a command file for most applications, there may be times when you want to create a special FASTTRAX.CM0. You may want to try your own ordering strategy or specify the exact order of the files and directories rather than use the placement priorities offered by MakeTrax.

Unless you have very special placement requirements, we suggest you skip this chapter.

You can create your special FASTTRAX.CM0 file with any text editor or word processor that supports the DOS text mode. To assist you, FastTrax can produce a complete list of the files and directories on your disk using the /O command line option.

The FASTTRAX.CM0 file consists of commands, file and directory names and comments. FastTrax looks only at the first four characters of each command name and accepts commands and filenames in upper or lower case letters.

The following paragraphs explain how to place your files and directories into *placement groups* as well as describe all the FASTTRAX.CM0 commands.

Placement Groups

With the FASTTRAX.CM0 file, you can divide the disk space into one or more groups. Each group has three parts:

o   Group command.
o   List of files and directories.
o   Free space command.

FastTrax makes default assumptions if one or two of these parts are not present. You can specify up to twenty groups; usually two is enough.

There are two group commands:

CONSECT   Define start of consecutive group.

The consecutive packing method assigns files and directories to consecutive clusters, producing contiguous files. However, the consecutive method alone is not as effective as the optimized method (described next) because it frequently produces cluster "orphans" -- file segments that needlessly span a cylinder boundary. The consecutive method can only be specified using the FASTTRAX.CM0 file.

OPTSECT   Define start of optimized group.

The optimized packing method assigns files and directories to consecutive clusters, but also attempts to assign each file and directory to the least number of disk cylinders. The optimized method minimizes cluster orphans and serves to further reduce track-to-track disk seeks. This is the normal (default) packing method used by FastTrax. If you omit the group, FastTrax assumes *OPTSECT*.

The list of files and directories, specified by their complete path name starting with a backslash (but not with a drive letter), defines the contents of the group. You can use the DOS wild cards *?* and * to specify a range of names within a directory, for example, \ABCD*.*. Use the *ALLOTHERS* command in the name list to show FastTrax where to place all unspecified names:

ALLOTHERS   All unspecified files will be placed here.

The FASTTRAX.CM0 file may contain at most one *ALLOTHERS* command; others are ignored. If no group contains *ALLOTHERS*, FastTrax places an *ALLOTHERS* group after the last group.

Within a name list, you can direct FastTrax to hold (not move) one or more names by bracketing these names with *HOLD* and *NOHOLD*:

HOLD   Hold all names until *NOHOLD*.

list of names

NOHOLD   (follows last name within list to be held)

To maintain compatibility with copy-protection schemes, FastTrax never moves files or directories with *hidden* or *system* attributes. You don't need to hold these items as FastTrax does that for you. As an extra precaution, you can hold all "visible" files that belong to a copy-protected program on the possibility that its protection scheme expects even these files to be in their original places.

There are two free space commands:

FREE n%   Assign n% of the free space here.

FREE   Assign remaining free space here.

If a *FREE* command is not present in either form, the group will not contain free space. If the FASTTRAX.CM0 file contains more than one *FREE* command (without n%), FastTrax divides the remaining free space among the *FREE* commands. If no group has a *FREE* command, FastTrax places the free space after the last group. For suggestions on how to assign free space, See *Assigning Free Space* in *Appendix B*.

Moving *read-only* Files and Directories

By default, FastTrax does not move files and directories with the *read-only* attribute. However, you can direct FastTrax to move read-only items with the *MOVE R/O* command:

MOVE R/O   Override hold of read-only files.

This command must appear before the first group in the command file. (When creating FASTTRAX.CM0, MakeTrax includes this command if you selected *move R/O* on the assignment screen.)
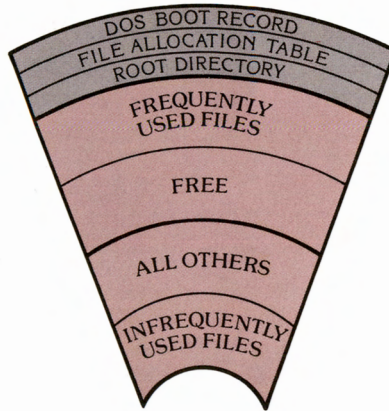
Comments in the FASTTRAX.CM0 File

You can include your own comments in the FASTTRAX.CM0 file. Since FastTrax expects commands and directory or file information to start at the first character on each line, you can include a comment by starting your comment with a space character in the first column. You can also place a comment at the end of a command by leaving a space between the FastTrax information and your comment.

FASTTRAX.CM0 Command Summary

Command	Meaning
ALLOTHERS	Place all unspecified files and directories here.
Comment	Start comment with a *space*.
CONSECT	Start a consecutive group. Follow with a list of files and directories.
FREE	Share remaining free space here.
FREE n%	Place n% of free space here.
HOLD	Hold all listed names until *NOHOLD*.
MOVE R/O	Move (don't hold) read-only files and directories.
NOHOLD	Ends *HOLD* range.
OPTSECT	Start an optimized group. Follow with a list of files and directories.
\Directory	Place directory here.
\Filename	Place file here.

FASTTRAX.CM0 Examples

Example 1: This example places a few frequently used files near the FAT and the rest at the high end of the partition. The free space is in the middle. The example also places two files near the very end of the partition:

The FASTTRAX.CM0 file contains:

OPTSECT	beginning of first optimized group
\COMMAND.COM	
\AUTOEXEC.BAT	
\WORDPERF	a directory
\WORDPERF*.EXE	all executable files in the directory
\WORDPERF*.FIL	the WordPerfect data files
\WORDPERF\TH.WP	the thesaurus
FREE	free space
OPTSECT	beginning of second optimized group
ALLOTHERS	all remaining directories and files not listed in either group
\config.sys	place at high end (Note: Same as \CONFIG.SYS, upper/lower case does not matter.)
\mode.com	place at high end

Since the last two filenames are at the end of the last group, FastTrax will place them at the high end or very near the high end of the disk partition. They may not be placed exactly at the high end because the optimized packing method may slightly alter the file order to achieve in-cylinder packing (see *Packing Methods* in *Appendix B*). Use the *CONSECT* group command if you need to specify the exact placement order.

Example 2: This example places some frequently used files near the FAT, followed by a free space, and the remaining files near the middle of the partition, followed by a second, smaller free space. Also, the example directs FastTrax to move read-only files and hold the files in two directories.
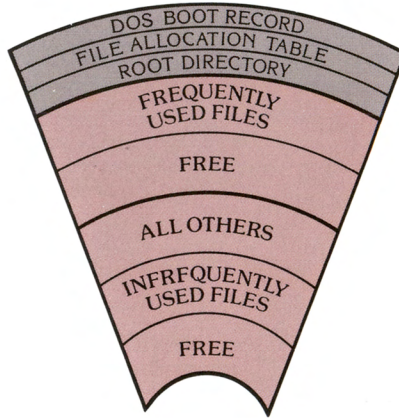
The FASTTRAX.CM0 file contains:

MOVE R/O	move read-only files
HOLD	hold names until *NOHOLD*
\DIR1	hold \DIR1 directory
\DIR1*.*	hold all files in \DIR1
\DIR2	hold \DIR2 directory
\DIR2*.EXE	hold .EXE files in \DIR2
NOHOLD	end of hold group
OPTSECT	beginning of first optimized group
\COMMAND.COM	
\WS4	place \WS4 directory near the FAT
\WS4*.EXE	place executable and overlay files next
\WS4*.OV?	
FREE 60%	some free space here
OPTSECT	beginning of second optimized group
ALLOTHERS	all other files here
FREE	remaining free space here

Appendix A: Questions & Answers

Q. How long will FastTrax take to optimize my disk?

A. That depends on many factors, including how much data must be moved, and the storage capacity and speed of both your disk and your computer. On an IBM AT 20MB drive, FastTrax usually takes less than fifteen minutes the first time; after that, usually less than ten minutes, and much less with the *fragmented files only* option. Typical optimizing times range between one and three megabytes per minute.

FastTrax displays the remaining bytes to move and the elapsed time to help you estimate when it will finish. Be aware that FastTrax optimizes faster at the beginning of Phase 3 and gradually slows down as it nears completion. You can always stop the optimization process (if you need your computer) by pressing the space bar or the *Esc* key, and resume at a later time.

Q. How often should I run FastTrax?

A. As often as you wish. Most users will run it once a week or once every two weeks. With the *fragmented files only* option, you can quickly optimize your disk at any time.

Q. On the fragmentation report, what is the difference between *optimized* and *consecutive*?

A. *Optimized* files reside on the least possible number of disk cylinders. Your disk requires time to move from one cylinder to another, so your computer can read a file more quickly if it resides on fewer cylinders. *Consecutive* files reside on contiguous clusters, but span one cylinder too many. In short, *consecutive* is good, but *optimized* is better.

For example, suppose a file requires ten clusters and each cylinder holds five clusters. If the file starts on the first cluster of a cylinder, then it spans two cylinders and is *optimized*. However, if the file starts in the middle of a cylinder, it spans three cylinders and is *consecutive*.

For more details, see *Packing Methods* in *Appendix B*.

Q. Why does the fragmentation map seem to show more fragmented files than is indicated by the fragmented percentage?

A. The map shows disk space, whereas the percentages are based on file counts. Also, the longer a file, the more likely it will be fragmented (a one-cluster file is never fragmented). Thus a small number of fragmented files may take up a lot of disk space. Another factor is that each map symbol often represents many clusters, and the symbol shown is based on the priority table (see *The Fragmentation Report*).

Q. When I use the optimized method, why does FastTrax leave some files as consecutive or fragmented?

A. The reasons are described generally under *Packing Methods* in *Appendix B*. Here's a specific example: Suppose each cylinder holds five clusters, and suppose also that FastTrax has assigned all but the last cluster of a particular cylinder. To be optimized, a file starting at that cluster must have a length of one cluster or six clusters or eleven clusters (and so on). If no such file is available, and since FastTrax is "packing" the disk, FastTrax is forced to make a *consecutive* assignment. Further, if the first cluster of the next cylinder is bad, and no one-cluster file is available, FastTrax is forced to make a *fragmented* assignment.

Compared with other disk optimizers, FastTrax leaves very few consecutive and fragmented files. (Some optimizers report a file that spans a bad cluster as *optimized*; FastTrax reports it as *fragmented*.) To simulate what other optimizers might produce, run FastTrax with a FASTTRAX.CM0 file that contains only a *CONSECT* command, and see the difference.

Q. I ran the DOS CHKDSK command with the /V switch to get a list of non-contiguous files. It showed none, but FastTrax reported that some files were fragmented. Why is this?

A. FastTrax includes fragmented directories in its report; CHKDSK does not. Note also that you must run CHKDSK (with the /V switch) on each directory to get a complete list of the non-contiguous files.

Q. Can I run FastTrax from a batch file?

A. Yes. Run FastTrax with the /B option from a batch file or the DOS command line. Include all non-default responses following the /B.

For more information on batch operation, see *Running FastTrax From a Batch File*.

Q. What else can I do to improve disk performance?

A. If you have not already done so, add a BUFFERS=n command (experiment using values of n between 4 and 30) to the CONFIG.SYS file in the root directory of your system disk. If you do not have a CONFIG.SYS file, create one containing the BUFFERS command. This command gives DOS more buffer space for retaining disk data, and the buffers improve disk performance by eliminating disk accesses. While this command has nothing to do with FastTrax, we mention it because some people do not know about it. See your DOS manual for more details.

Q. How does optimization affect un-erasing a file?

A. If you erase a file and then run FastTrax, you will not be able to recover it because FastTrax removes all deleted directory entries. One of the purposes of FastTrax is to reclaim directory space.

However, after running FastTrax, if you accidentally erase a file, the chance of recovering the file immediately following the erasure is very good. It is much easier to recover an erased file with consecutive clusters.

Q. I use SpeedStor (from Storage Dimensions, Inc.) to partition my disk. FastTrax and MakeTrax work fine on the first partition, but they report "Inconsistent data" on the remaining partitions. However, CHKDSK reports no errors. What's wrong?

A. With SpeedStor you must initialize the disk, define the partitions, and then format each partition with the SpeedStor PartEd program. For safety, FastTrax and MakeTrax will not operate unless the partitions are properly formatted.

Back up the partitions, run PartEd to format the partitions, and then restore the partitions. FastTrax and MakeTrax should then run fine.

Q. After running FastTrax, why do the directories look the same?

A. File placement and directory order are not the same thing. Although some optimizers change the directory order, FastTrax does not, viewing this as an unrelated action.

You can confirm that FastTrax has properly moved your files and directories by invoking MakeTrax and checking the file and directory placement with the file map window.

Q. After running FastTrax, some programs run more slowly. Why is this?

A. With the default order (no \FASTTRAX.CM0 file), FastTrax may move a file that was originally near the FAT to a less optimum place.

Since all files cannot be near the FAT, you can use MakeTrax to define which files and directories merit faster access.

Q. My disk has several very large files. After running FastTrax, some of these files are still fragmented. Why?

A. If your disk has bad clusters or clusters that contain data that should not be moved, FastTrax may not be able to find a single contiguous disk area for a very large file, in which case it will assign the file across one or more bad or hidden clusters.

Unlike many disk optimizers, FastTrax accurately depicts a file that straddles a bad or hidden cluster as fragmented. However, for large files this representation looks much worse than it actually is. In fact, occasional intervening clusters in a very large file have little effect on performance.

To confirm that the file is essentially optimized, invoke MakeTrax and use the disk map to examine the placement of these fragmented files. In most cases, the files will reside in a few disk areas and span no additional cylinders.

Q. Some disk optimizers place all directories near the FAT. Can I do this with FastTrax?

A. FastTrax offers flexibility -- you can place them anywhere you wish. If you want your directories near the FAT, you can place them there with MakeTrax -- see the last example under *Examples of Group Priority Assignments*. By default, FastTrax assigns files near their directories. This method works best for programs that reference files and their directories at roughly similar rates, which is the case for most programs. See *The Default Packing Order* in *Appendix B* for more details.

Q. Can I optimize a RAM disk?

A. Yes. Since a RAM disk has no moving parts, in-cylinder optimizing does not apply. On the fragmentation map, FastTrax identifies unfragmented files on a RAM disk as *contiguous* (rather than *optimized* or *consecutive*).

While FastTrax is designed to optimize mechanical disks, there are certain advantages to optimizing RAM disks. First, it's much easier to recover a contiguous file after accidental erasure. In addition, DOS can search directories more quickly when there are no deleted entries, and DOS can use less buffers when files are near their directories (see *The Default Packing Order* in *Appendix B*). Finally, DOS can load a contiguous file in just one disk read whereas a fragmented file requires at least two disk reads. Since you may have to reboot after optimizing, you may only want to optimize a RAM disk that has a battery backup.

Q. Why does FastTrax have such a plain user interface?

A. FastTrax has a simple interface to maximize memory for internal tables. The MakeTrax utility has different memory requirements, so we could afford to implement a more elaborate user interface with color and help screens.

Many disk optimizers cannot handle large disks or disks with many files, directories, directory levels, and clusters. FastTrax can optimize very large disks (the present DOS limit is 4,000 megabytes), but the extreme cases require a lot of memory. By keeping the user interface simple, FastTrax can handle larger cases on systems with no expanded memory. But, in response to comments from users, we may develop an enhanced interface for use on systems with expanded memory.

Appendix B: About Disk Optimizing

Definitions

Cluster. The basic unit of data storage for DOS is the cluster. DOS assigns the cluster size during the formatting process. A common cluster size on IBM PC and compatible hard disks is 2K (2,048 bytes). A cluster consists of one or more sectors, and the number of sectors in a cluster must be a power of two. Some clusters may span cylinder boundaries. Whether they do or not depends on the number of sectors per track, tracks per cylinder, and the cluster size.
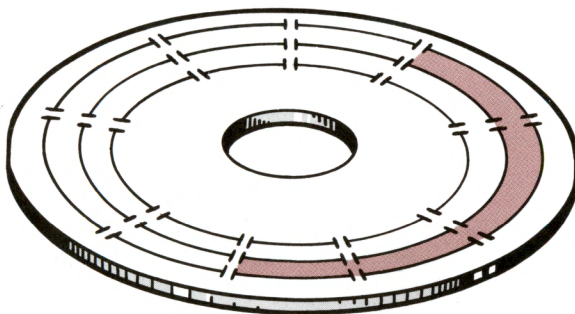
CLUSTER

Contiguous File. All clusters are in numerical sequence. DOS looks at the disk as a sequence of sectors and clusters. The FastTrax consecutive packing method organizes the files and directories as contiguous. This method will improve hard disk performance, but it can be improved upon with the FastTrax optimized packing method. The FastTrax optimized packing method also makes each file a contiguous file, but additionally, it attempts to keep files within the least number of cylinders to minimize track-to-track head movement.

Cylinder. A cylinder is a collection of all tracks that are located the same distance from the outside edge of all the platters on the drive. This allows the gang of read/write heads to access multiple tracks without any stepping motion.
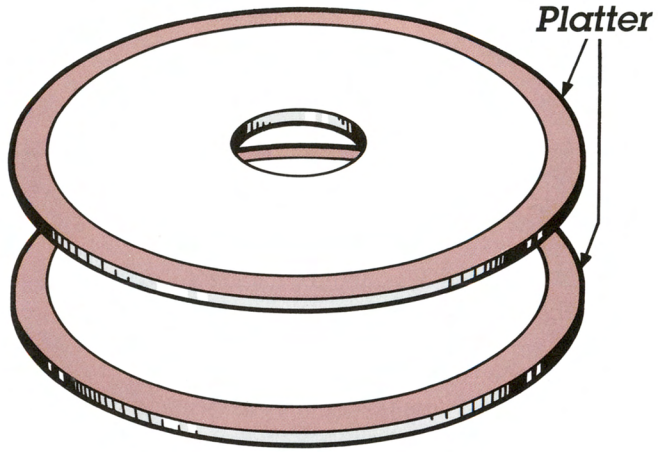
CYLINDER

Fragmented File. A fragmented or non-contiguous file is a file that is larger than one cluster and has one or more clusters that are numerically separated from other clusters. The following illustration depicts several fragmented files on a disk.
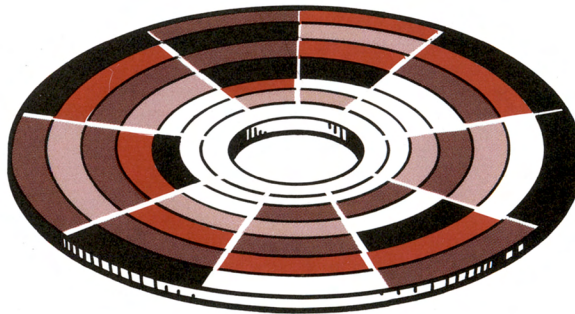
FRAGMENTED DISK

Hard Disk Drive. The hard disk drive used on most personal computers is a Winchester-technology hard disk. This is a disk unit that has one or more platters and read/write heads in a sealed environment. The platters are either fixed into the drive or part of a removable unit. The fixed unit is the most common; it consists of one or more platters, one or more read/write heads per recording surface, some motors, and some electronic circuitry.

Interleave. Although sectors are logically in numerical order on a track, they may or may not be physically next to each other. Interleave is a scheme that allows enough time for the computer to read or write consecutive sectors without losing a disk revolution between sectors. The interleave factor minus one is the number of sectors the computer will skip before it reads or writes the next sector. FastTrax considers interleave as it optimizes.

Partition. DOS allows a hard disk to be divided into several contiguous areas called partitions. Using the DOS FDISK command, you can create these partitions and direct DOS to use some or all as "logical disk drives", each with its own drive letter. Remaining partitions can be used by other operating systems, such as UNIX.

Platter. The platter is a rigid non-magnetic metal (generally aluminum) disk, coated with a thin plating of a magnetic material (typically ferric oxide). Usually both sides of the platter are coated. The following illustration shows a pair of platters with four recording surfaces.

Side 0

Side 1

Side 2

Side 3
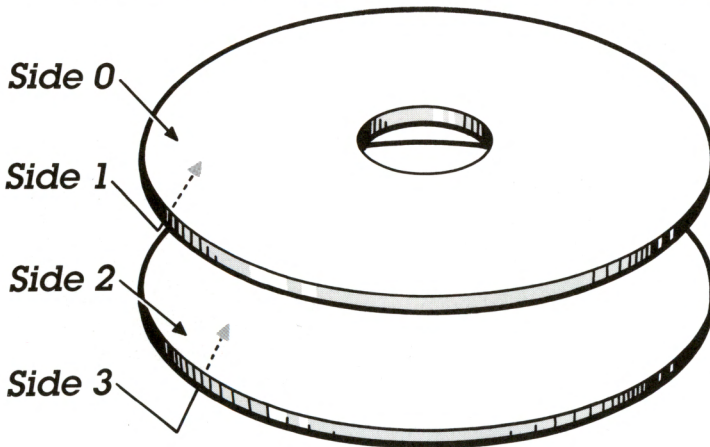
TWO PLATTERS

Sector. The basic unit of data storage on a disk is a sector. A sector is typically 512 bytes.
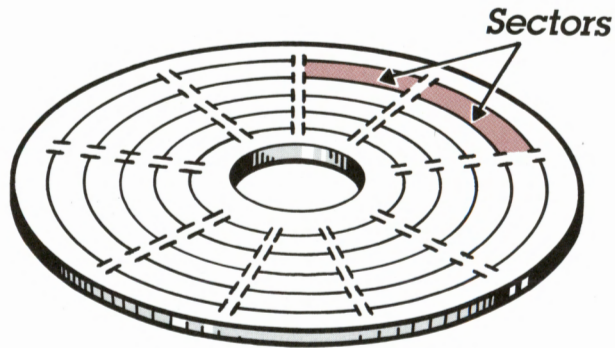
SECTORS

Track. Each recording surface of a platter is divided into concentric rings called tracks. Each track is divided into a number of sectors during the low level formatting. The following illustration shows a platter with several tracks.
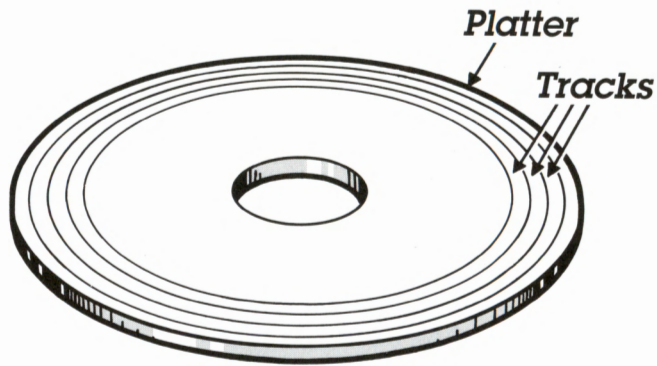
PLATTER WITH TRACKS

The Optimizing Process

The "optimizing process" used by FastTrax consists of four phases:

Phase 1   Removes deleted directory entries and reclaims unused directory clusters. (Phase 1 does not remove empty directories.)

When a file is erased, DOS does not delete the file's directory entry, but only marks the entry as "deleted" so it can be used by a new file entry. By removing deleted directory entries, FastTrax lets DOS search your directories more quickly.

Phase 1 removes deleted entries by "compressing" each directory. Then, if the last cluster (or clusters) of some directories are not needed, Phase 1 releases them back to DOS.

Phase 2   Determines the new organization of your files, directories, and free space, either using the FASTTRAX.CM0 file or the default ordering method.

Phase 3   Reorganizes your disk.

This phase usually requires more time than the other phases. Typical optimizing times range between one and three megabytes per minute.

FastTrax periodically displays the remaining bytes to move and the elapsed time to help you estimate when it will finish.

You can abort this phase at any time by pressing the space bar or the *Esc* key, and then resume the process later by restarting FastTrax. If you don't add or delete any files, FastTrax will continue exactly where it left off with little time loss. It is not necessary to use the recovery procedure after aborting, because FastTrax completes Phase 4 before exiting.

After aborting, the files may appear to be more fragmented than they were at the outset, but they are actually closer to the placement order determined in Phase 2.

Phase 4   Completes all deferred disk writes and asks if you would like to see the new organization. A *no* answer will return you to the opening menu (or to DOS if you specified a drive letter on the command line). A *yes* answer will display the new fragmentation report.

Packing Methods

FastTrax assigns files and directories either in directory order or in the order specified in the FASTTRAX.CM0 file. With directory order, FastTrax assigns all entries in a given directory physically near the directory itself in the order the entries appear in that directory. With FASTTRAX.CM0 file order, FastTrax follows an optional user-supplied text file described under *The FastTrax Command File*.

FastTrax offers two packing methods that move files and directories to more efficient places on a disk:

o   Consecutive Packing Method. This method assigns files and directories to consecutive clusters, producing contiguous files. This is sometimes called *de-fragmenting* and it is a good technique for increasing disk access speed. However, the consecutive method alone is not as effective as the optimized method (described next) because it frequently produces cluster "orphans" -- file segments that needlessly span a cylinder boundary. The consecutive method can only be specified via the FASTTRAX.CM0 file.
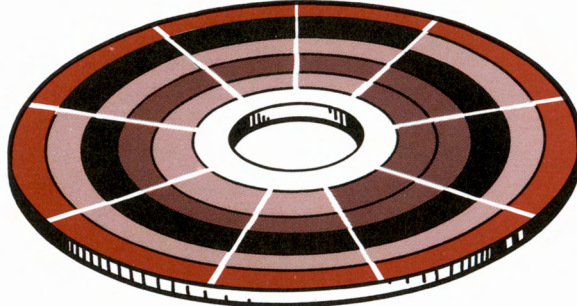
CONSECUTIVE PACKING

o   Optimized Packing Method. With this method, FastTrax performs *de-fragmentation* as described in the consecutive method, but also attempts to assign each file and directory to the least number of disk cylinders. The optimized method minimizes cluster orphans, thus reducing track-to-track disk seeks. This is the normal (default) packing method used by FastTrax; it may also be specified in the FASTTRAX.CM0 file.
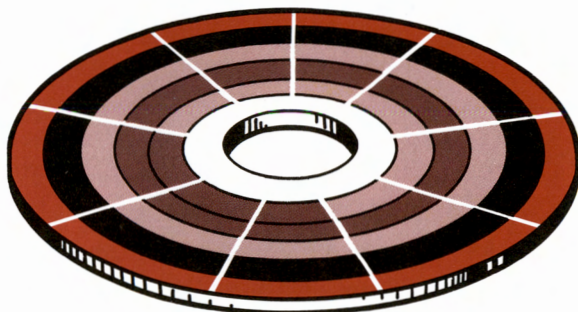
OPTIMIZED PACKING

You may never need the consecutive method because the optimized method is better. One possible use: If you have an application program that runs faster when its files and directories are in a specific order on your disk, you can use the consecutive method in conjunction with the FASTTRAX.CM0 file to produce this order.

FastTrax may not be able to use optimized packing for certain files, even if you've specified the optimized method. (You'll know if this has happened, because the fragmentation report will show a few consecutive-packed files.)

With the optimized packing method, FastTrax will, if necessary, deviate slightly from the default or FASTTRAX.CM0 order to assign files to the least number of cylinders. However, in some cases, even when it deviates slightly from the indicated file order, FastTrax cannot successfully utilize the optimized packing method. In these cases it will revert to the consecutive packing method to maintain the indicated file order. And finally, FastTrax sometimes cannot use optimized packing for the very last file (the file left after all other files are packed) if that file spans a cylinder boundary.

With either the consecutive or the optimized packing method, FastTrax may encounter bad clusters and held clusters when assigning a file (or directory). Although FastTrax bypasses these clusters, it may assign a file around them, resulting in a fragmented file. Fragmentation caused by intervening bad clusters and held clusters happens much less often with the optimized method because FastTrax alters the file order to avoid fragmentation.

The Default Packing Order

By default, FastTrax assigns files near their directories. This method works best for programs that reference files and their directories at roughly similar rates, which is the case for most programs.

Below we explain why the FastTrax method allows DOS to use fewer buffers when referencing files and their directories. DOS then has more buffers for reducing file and directory accesses.

To access a file, DOS must read the file's directory, the file itself, and the FAT. For minimum disk seeks, all three structures must be near each other. But this is not possible for most files because there is one FAT and its location is fixed.

The next best approach is for two of the structures to be near each other. There are three cases:

1) The directory is near the FAT (and the file is not).

   To access the file, DOS must first search the directory. To search the directory, DOS must read a portion of the FAT into a DOS 512-byte disk buffer. To read the file, DOS must read a different portion of the FAT into a second disk buffer.

   In this case, DOS usually requires two disk buffers to hold FAT information. (DOS also uses other disk buffers to read the directory and file.)

2) The file is near the FAT (and the directory is not).

   The comments for Case 1 also apply to Case 2.

3) The directory is near the file (the FastTrax default method), and neither are near the FAT.

   This case is similar to the prior cases except that DOS may only need one disk buffer to hold FAT information.

   Here's why: With a 16-bit FAT, a 512-byte DOS buffer holds 256 FAT entries. Even with "small" 2,048 byte clusters, one DOS FAT buffer controls a half million bytes of storage. The chance is very high that both the directory and the file are under the control of a

single FAT buffer. The chance for this is even higher with larger clusters.

Freeing one disk buffer increases the chance that the DOS disk buffers will eliminate disk reads.

The Case 1 and Case 2 approaches have merits for certain types of programs and applications. With MakeTrax, you can use these approaches by assigning differing placement priorities to certain files and their directories.

The Case 1 approach is useful when many directories are searched before a file is found. For example, when DOS searches the DOS PATH for a particular program, DOS will find the program more quickly when the directories on the DOS PATH are placed near each other and also near the FAT. In this case, the placement of the individual programs is less important.

The Case 2 approach is useful when files are referenced much more often than their directories. For example, if a database program references its files very often without creating new files, the program will run faster if the files are placed near each other and also near the FAT. In this case, the placement of the directory is less important.

Assigning Free Space

For some applications, it may seem reasonable to assign a free area after each frequently updated file (or group of files) in the hope that DOS will place file updates in a nearby free area. Unfortunately, DOS takes a different tack.

DOS uses two different free space allocation methods, depending on the version of DOS:

DOS 2.xx always allocates free clusters starting nearest the FAT.

DOS 3.xx and 4.xx use a wraparound method; they start with the free clusters nearest to the FAT, and then continue forward (farther away from the FAT) until the end of the partition is reached, even if clusters closer to the FAT become available. (With DOS 3.xx and 4.xx, you have a much better chance of "un-erasing" a deleted file.)

Given the way DOS allocates free space, one *FREE* area is best for many applications. Two *FREE* areas are useful if you won't need the space in the second area before you again optimize your disk. For DOS 2.xx, that may be a long time because DOS 2.xx always allocates from the low end. Whether or not two *FREE* areas are useful under DOS 3.xx and 4.xx depends on how long you use your computer before rebooting or shutting it off.

Placing infrequently used free space at the high end of a partition is similar to placing infrequently used files there -- the average seek time for the disk partition is lower because you don't use the upper clusters.

Appendix C: Configuring FastTrax

FastTrax and MakeTrax will work properly without any special configuration procedures on most IBM and IBM-compatible computers and disk drives.

In a small number of cases, FastTrax and MakeTrax may incorrectly determine the "geometry" of your disk drive. In these cases, FastTrax will operate safely, but it may not perform the "in cylinder" optimization correctly. Although most files will be consecutive, some files may cross cylinder boundaries even though FastTrax reports them as *optimized*. Your disk performance will be good, but it could be better. MakeTrax will also operate safely, but it will incorrectly show some files and directories as *optimized* when they are actually *consecutive* and vice versa.

To perform "in cylinder" optimization, FastTrax and MakeTrax must know four parameters about your disk drive:

Parameter	Meaning
heads	number of read/write heads per cylinder
sectors	number of physical sectors per track
sector size	number of bytes in a physical sector (usually 512 bytes)
offset	number of physical sectors in the first cylinder of the DOS partition below DOS sector 0

FastTrax makes a concerted attempt to determine these parameters. But if your drive or controller emulates another drive or uses some type of "sector translation", FastTrax may not recognize the drive's true parameters.

To list FastTrax's determination of your drive's parameters, type:

FASTTRAX d: /P   (where *d:* is a drive letter)

FastTrax displays the four parameters in the order shown above.

If you know the parameters to be incorrect, you can direct FastTrax (and MakeTrax) to use the correct parameters by creating a FASTTRAX.PAR file in the root directory of the drive. The FASTTRAX.PAR file must contain a single line starting with *DRIVE* and followed by the four parameters. For example, if you know the correct parameters to be 5 heads, 17 sectors per track, 512 bytes per sector, and 17 offset sectors, respectively, then your FASTTRAX.PAR file should contain:

DRIVE 5 17 512 17

If you are unsure of the offset value, specify *-1* for this parameter and FastTrax will calculate it. This is particularly useful for a drive or controller that employs emulation or sector translation because the offset can be difficult to calculate. Similarly, you may specify *-1* for the sector size and FastTrax will calculate it. For example, if you know your drive has 15 heads, 36 sectors per track, and 512 bytes per sector, but you are unsure of the offset, your FASTTRAX.PAR file should contain:

DRIVE 15 36 512 -1

To create the FASTTRAX.PAR file, use a text editor or word processor that supports the DOS text mode.

Appendix D: Recovery

Overview

If your computer system fails (e.g., power failure) while running FastTrax, the chance of losing data is very small. During file reorganization, FastTrax maintains file integrity if the failure occurs between disk writes or when writing to any data cluster.

However, FastTrax may not maintain file integrity if a system failure occurs while writing to the File Allocation Table (FAT) or any directory. This is a DOS limitation and applies to all programs that run in the DOS environment, including DOS itself and other disk optimizers, because DOS maintains a single copy of the directories. On most disk media, DOS maintains two copies of the FAT, but (as yet) does not make use of the second FAT. In effect, there is one copy of the FAT and it is as vulnerable as the directories.

On most systems, a system failure is a rare event, and since most FastTrax disk writes involve data clusters, the chance of preserving file integrity after a system failure is very high. To gain speed, FastTrax defers certain disk writes until the final phase of the program. However, FastTrax always writes enough information to the disk to allow CHKDSK to fully restore file integrity in the event of a recoverable system failure.

Before restoring file integrity, CHKDSK may output many exception messages. If FastTrax has not completed Phase 4, these messages are normal, and full file recovery is highly probable, in spite of the messages.

How to Recover Your Data

This section gives step-by-step instructions on how to recover your data after a system failure or after finding a strange directory listing.

1. Reboot your system (but read the next paragraph first).

    It's very important that no program write to the target disk during the reboot process. If you are sure that your AUTOEXEC.BAT does not

contain any command that may write to the target disk (such as a COPY command), reboot your computer system. Otherwise, reboot using a floppy disk that has no AUTOEXEC.BAT file.

2. Use the DOS CHKDSK command to look for disk errors:

    CHKDSK d: (where *d:* is the target drive)

    If CHKDSK reports no errors, your disk should be okay. You may have been running FastTrax with a terminate-and-stay- resident program (such as DOS's FASTOPEN program) that did not follow the changes FastTrax made to your disk.

    Even if CHKDSK outputs many error messages, your data is probably okay.

    CHKDSK may output many "invalid directory entry" messages. Don't worry about these messages.

    CHKDSK may also output:

        xxx lost clusters in yyy chains.
        Convert lost chains to files (Y/N)? (answer N for now)

    Don't worry about this message either.

    If these are the only errors found by CHKDSK, proceed to Step 4, where you should be able to recover your disk with no data loss.

3. If CHKDSK lists other errors, the system failure corrupted the root directory, a subdirectory, or the FAT, and some or all of the data may not be recoverable. If you call our Technical Support number, we may be able to help you. If you have a good backup, then:

    a) Run CHKDSK with the */F* option to clear the error messages. If errors persist, you may have to reformat your disk.

    b) Restore your disk from your backup media.

    c) Please call us if you think FastTrax caused or contributed to the data loss.

4. Run CHKDSK with the */F* switch and answer No to the "Convert lost..." question. The lost clusters contain some original parts of your files; since these parts have been copied successfully to their new destinations, they are no longer needed.

Appendix E: Error Messages

This section explains the FastTrax error messages, listing them alphabetically.

Disk error while reading sector *n* from drive *x: ...*"
Disk error while writing sector *n* to drive *x:*

FastTrax detected an error reading or writing a sector on your drive. Even so, the data on the remainder of your disk is probably okay (especially after a read error). Have your hardware checked before running CHKDSK with the */F* parameter. Then follow the FastTrax recovery procedure. Note that CHKDSK checks only the directories and the File Allocation Table (FAT); it does not check each cluster on the disk. Therefore, it's possible that your disk has more bad clusters even though CHKDSK shows no problems.

Disk is full - FastTrax needs at least 1 free cluster

To optimize safely, FastTrax needs at least one free area of disk during optimization. To run FastTrax, delete a file that has a non-zero byte count.

Drive *x:* is not ready

DOS recognizes the drive as valid but cannot access it, possibly because the drive is not turned on, or the drive door is open, or the media is missing. Correct the problem and continue. You may see this message if you specify a network drive while the network is active. You should disable all networks before running FastTrax.

Drive *x:* is write protected

This may occur when you are using a removable media disk. These disks usually have a tab or some other device that can be set to prevent a write to that disk. The write protection must be removed before FastTrax will work on that disk.

Duplicate name: *filename*

The filename appeared earlier in the FASTTRAX.CM0 file. FastTrax uses the first instance of the name to determine file placement and ignores duplicate names.

Error during batch operation

FastTrax detected an error during batch operation, and because you specified the *pause on error* batch option, FastTrax paused before returning to DOS to allow you to examine the error.

Error in SET statement

There is some error in your SET FT_EMSDRV=x: statement. Make sure you typed it correctly, without any extra spaces.

Expanded memory error

FastTrax found an error in expanded memory, possibly caused by a bad memory chip. Have your computer serviced.

FastTrax found this problem ...

FastTrax found a problem after optimizing your disk. (FastTrax describes the problem prior to displaying this message.) Since FastTrax confirmed that your disk was okay before optimizing, the problem was probably caused by some sort of failure in your system. Some data may have been lost. To prevent further loss, avoid any action that will modify your disk. You'll probably have to restore your disk from your latest backup. Call our Technical Support number and we may be able to help you recover your data.

FREE percent exceeds 100

The total of the *FREE n%* commands in the FASTTRAX.CM0 file exceeds 100 percent. Correct your FASTTRAX.CM0 file and continue.

Held name: *filename*

FastTrax will not move the indicated file or directory because it has a *hidden*, *read only*, or *system* file attribute. This message is just a notification; it doesn't stop the optimizing process. You can move read-only files and directories if you include the *MOVE R/O* command in the FASTTRAX.CM0 file.

Inconsistent data ...

While analyzing your disk (and before taking any action that might alter your disk), FastTrax found a discrepancy between the data in the directories and the data in the FAT.

To correct this problem, check the integrity of your disk with the DOS CHKDSK command. If CHKDSK finds errors, correct them according to the directions in your DOS manual and run FastTrax again. If CHKDSK finds cross-linked file pairs, copy each file to a new place, delete the original files, and run CHKDSK again with the /F switch. Then try to examine the files. At least one will have inappropriate data starting from the end of the file. If possible, correct the files with an editor or disk utility. Otherwise, restore the files from your latest backup.

Invalid command: *text*

The indicated text is not a valid FASTTRAX.CM0 file command.

Invalid drive letter: *x:*

DOS does not recognize the drive letter.

Invalid name: *name*

The indicated name is invalid. Perhaps it's a directory name that contains wild card characters; wild card characters may only be used to specify a group of filenames.

Invalid parameter: *parameter*

The indicated parameter is not allowed on the DOS command line. See *Running FastTrax* for a description of the allowable syntax.

Memory error

FastTrax found an error in low memory, possibly caused by a bad memory chip. Have your computer serviced.

Not enough RAM to continue

This happens when the target disk has too many files and directories for FastTrax to manage in the available memory. If possible, reconfigure DOS with more available RAM by eliminating resident programs and virtual disks. If you still get this message after reducing the size of your DOS, then you need expanded memory (LIM 3.2 or higher) to optimize your disk.

If you are using an EMS driver that emulates expanded memory on a hard disk (such as Above Disc from Above Software, Inc.), you must use the *E* option to specify the drive letter of the emulated disk. See *Command Line Options* for more details.

Not found: *name*

The indicated file or directory name in the FASTTRAX.CM0 file is not on the target disk. This message is just a notification; it doesn't stop the optimizing process. You probably deleted the file or directory after you created FASTTRAX.CM0.

Too many directory levels

Your disk has more than 33 directory levels. This is an unusual situation because DOS path names may not exceed 64 characters, which in turn limits the maximum number of directory levels to 31.

Too many groups

The FASTTRAX.CM0 file contains more than 20 *OPTSECT* and *CONSECT* groups. If you need more than 20 groups, please write and we will try to help you.

Unable to read drive *x:*

FastTrax cannot read the FAT. Your disk may be faulty or is under the control of a non-standard disk driver. FastTrax cannot optimize this disk.

Unsafe output redirection

You can use the DOS output redirection feature '>' with many FastTrax command line options. But you cannot redirect output to the disk you are optimizing.

Appendix F: In Case of Difficulty

If you are having trouble using FastTrax:

1. To find out if a resident program is interfering with FastTrax, try running FastTrax from a floppy-based "minimal" DOS. See *Installing FastTrax* for instructions on how to make this floppy disk. Then, if FastTrax runs properly, add resident programs one at a time to find the incompatible program.

2. Check the integrity of your FastTrax .EXE files by running the FASTINFO program from the directory where you keep FastTrax. If FASTINFO finds corrupted files, restore them from your master FastTrax disk.

3. Check your hardware by running your computer's diagnostic programs. Also, look for an error message when your computer first starts. (Some computers have a RESET button that you can use instead of turning the power off and then on.)

If you are unable to locate a problem, please jot down the answers to the questions below before contacting our Technical Service department:

1. What is your FastTrax version number?. FastTrax displays its version number on every screen.

2. Describe you computer hardware:
   a. name and model.
   b. RAM size.
   c. Expanded Memory size.
   d. add-on boards (if you're aware of any).

3. What hard disks are you using?
   a. brand and size.
   b. controller type (if you know it).

4. What is in your CONFIG.SYS file?

5. What is in your AUTOEXEC.BAT file?

FastTrax technical support is available Monday through Friday from 9:00 a.m. to 5:00 p.m., Pacific Time. Please call or write:

<div align="center">

FastTrax International
8 Digital Drive, Suite 102
Novato, CA 94949
(415) 883-9561

</div>

# Index

PC WORLD BEST BUY

"Being able to customize optimizations to fit your applications is what makes the difference . . . FastTrax wins hands down. It's the most fully featured optimizer around."

PC World, June 1988

PC MAGAZINE EDITOR'S CHOICE

"With more power and more options than the competition, and with all available safety precautions, FastTrax is the closest thing you can find to the ultimate disk packer and defragmenter."

PC Magazine, Oct. 11, 1988

1989 BYTE AWARD OF DISTINCTION
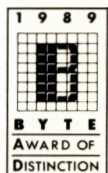
"But after trying them all, we've found FastTrax to be fast, rock-solid, easy to use, and highly configurable — your disk gets organized the way you want it in minutes."

Byte Magazine, January, 1989

What slows down your hard disk?

File Fragmentation—From the day you began using your hard disk, DOS was storing your files in pieces all over the disk—instead of keeping them in one place. This is called "fragmentation." It makes your hard disk seem to run much slower.

File Placement—DOS doesn't care very much where it puts the pieces of files that it fragments. When your files aren't located close to the File Allocation Table (FAT)—or if they're not completely within disk cylinder boundaries—your hard disk takes longer to access them.

And, it only gets slower—the more work you do, the more your files will become fragmented—unless you optimize your hard disk.

FASTTRAX speeds up your hard disk performance in 3 ways.

- **FASTTRAX** defragments your files and directories by consolidating the scattered pieces as much as possible into one location.

- **FASTTRAX** places most of your files within disk cylinder boundaries. This speeds up your hard disk performance even more by reducing time-consuming track-to-track seeks when accessing these files.

- **FASTTRAX** and its companion program, MakeTrax (included), give you the flexibility to place your most frequently used files on the "fast tracks" nearest to the FAT, reducing the amount of movement required of your hard disk's read/write head.

See the FASTTRAX difference.

You'll see results immediately with **FASTTRAX**. Your hard disk will perform "like-new," running smoother and faster—with less effort—than it has in a long time.

Get the FASTTRAX edge.

- Optimize any disk supported by DOS, including those with DOS partitions larger than 512 MB.

- Fast and safe optimizing.

- Extra-fast "optimize fragmented files only" option lets you "tune-up" your disk in seconds.

- Visual reporting shows fragmented disk areas and other file statistics before you optimize.

- No limitations on the number of files, directories, directory levels, or disk clusters.

- Batch file operation.

- Works on hard disks or floppies.

- Helps prolong the life of your hard disk.

- Compatible with existing copy protection schemes; FastTrax itself is not copy protected.

Put your system on the "fast tracks."

- **FASTTRAX** is the disk optimizer that speeds up performance, extends life, and reclaims wasted space.

System Requirements: Works on IBM PS/2, PC/XT/AT and compatibles using PC-DOS or MS-DOS version 2.0 or higher.

Minimum RAM requirements:
- 256 KB for floppy disk systems
- 512 KB for most hard disk systems

Maximum RAM requirements:
- 800 KB expanded memory (LIM version 3.2 or higher) may be needed to optimize hard disk partitions larger than 85 MB.

FASTTRAX INTERNATIONAL

8 Digital Drive, Suite 102, Novato, CA 94949
(415) 883-9561