



DFS

Dos File System

User's Guide

First Edition

*For QNX® Version 1.2 / 2.0
Release 2 or later*

Copyright 1989 Quantum Software Systems, Ltd.
ALL RIGHTS RESERVED.

Published by

Quantum Software Systems, Ltd.
Kanata South Business Park
175 Terrence Matthews Cr.
Kanata, Ontario K2M 1W8
Canada
(613) 591-0931

© 1989 Quantum Software Systems, Ltd.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without the prior written permission of Quantum Software Systems, Ltd. Printed in Canada.

First Printing March, 1987
Second Printing January, 1989

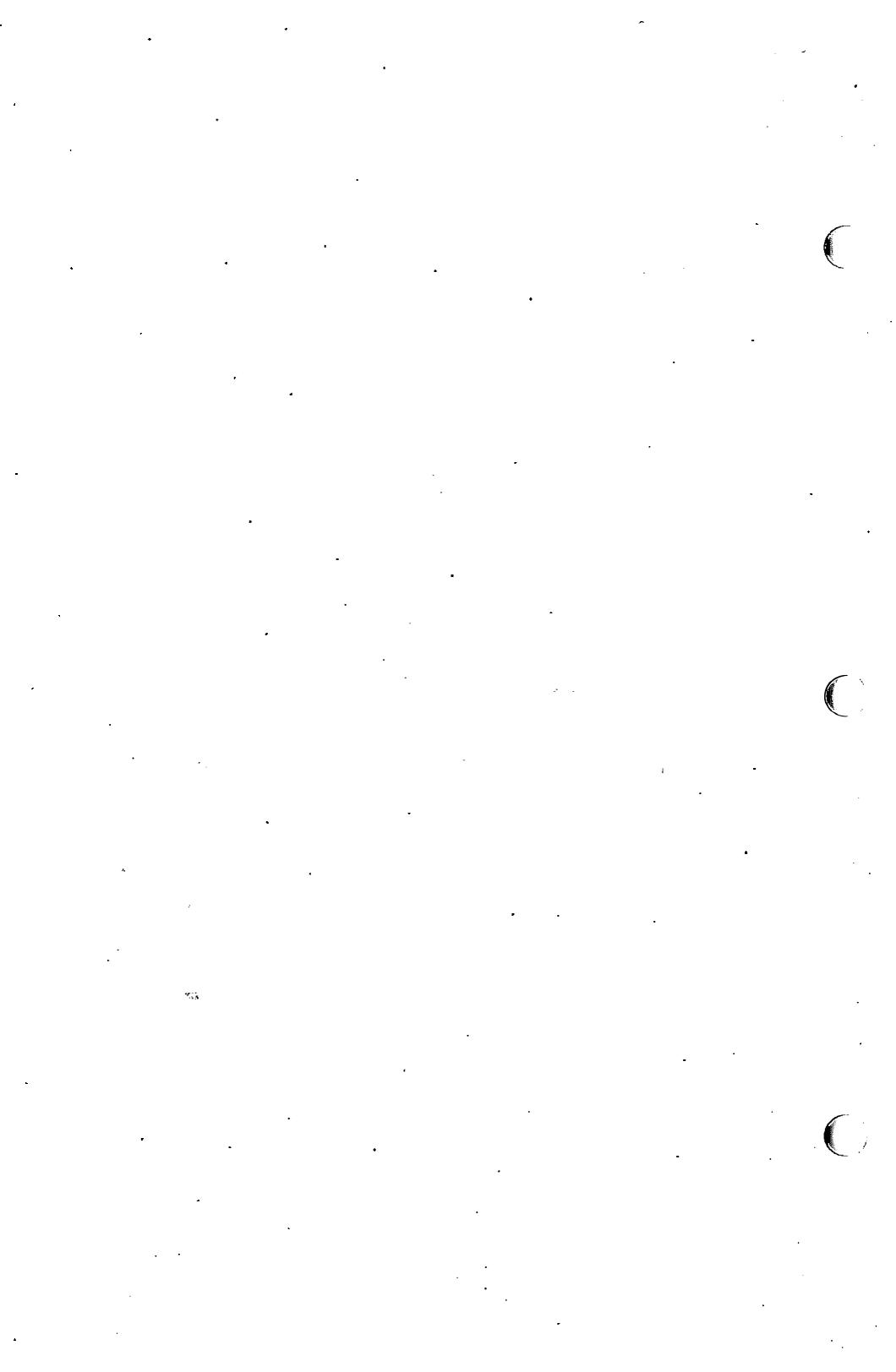
QNX is a trademark of Quantum Software Systems, Ltd.

DOS FILE SYSTEM

Page

1. The DOS File System

1.1	Introduction	1
1.2	DOS Text Files	1
1.3	Using the DOS File System	2
1.3.1	Description	2
1.3.2	Installation	3
1.3.3	Backing up DOS partitions with DFS	3
1.3.4	DOS-QNX and QNX-DOS character mapping	4
1.4	Command Syntax	5
1.5	Sample DFS Configurations	9
1.5.1	Example I	9
1.5.2	Example II	9



1. The DOS File System

1.1 Introduction

The DOS FILE SYSTEM (DFS) is a program which allows you to process files on DOS disks as if they were QNX files. You may create files on DOS disks, read them, write them and delete them with any QNX program. You may also create, read and delete directories on DOS disks with the standard QNX utilities such as MKDIR, RM, FILES, etc. All standard QNX utilities will work with DOS files, including such programs as ED.

Your own programs will also be able to process DOS files in the same way as QNX files using the standard QNX "C" I/O functions such as FOPEN, FGET, FPUT, FSEEK, etc. When you read DOS directories they will be presented to you in QNX format. The only limitation imposed on your programs is that you may not seek in directories or in DOS text files when they are being accessed as text files.

1.2 DOS Text Files

Text files are defined to be line-oriented files containing lines of ASCII text separated by line separator sequences. DOS uses a different structure for text files than QNX does. Consequently, it is important to be able to tell DFS how to recognize DOS text files so that the proper conversion between DOS and QNX format may be performed. All files which are not text files are called binary files.

In DOS, each line of a text file is terminated with a line feed/carriage return (LF/CR) sequence, while in QNX each line is terminated by a record separator (RS) character. In addition, some text files created by early DOS programs may contain a SUB character as the last character of the file.

DFS will automatically convert between QNX and DOS text file formats if it knows how to detect a DOS text file. The following mechanisms are provided to allow you to control the detection of DOS text files.

1. You may specify that all files are to be treated as text files.
2. You may specify that all files are to be treated as binary files.
3. You may specify that files with certain file extensions are to be treated as text files and that files with other file extensions are to be treated as binary files.
4. If you provide lists of text and/or binary file extensions, you may specify whether files with unrecognized file extensions are to be treated as text or binary files.

5. An individual file may be forced to be treated as a text file by adding the suffix "**~t**". This will override any other file-type recognition rule.

ed a:/file.abc~t - force file to be treated as a text file.

6. An individual file may be forced to be treated as a binary file by adding the suffix "**~b**". This will override any other file-type recognition rule.

copy a:/file.xyz~b 3:/dos - force file to be treated as a binary file.

1.3 Using the DOS File System

1.3.1 Description

DFS contains two components, a command interpreter, which allows you to define and alter the environment of DFS, and the DFS administrator, which provides the DOS file system services during program execution.

The command interpreter provides the following services:

- **Define initial DOS file system environment at start-up;**
- **Modify DOS file system environment at any time during execution;**
- **Display the current DOS file system environment;**
- **Stop the DOS file system;**

The DOS file system environment is made up of the following elements which control the operation of the DOS file system.

DOS drives.

You may define up to 6 DOS drives (a - f) and map these to QNX drives which have been mounted with the standard **QNX MOUNT** command. You may also associate DOS drives with pseudo-disks which are used with the QDOS product. As part of specifying the association between DOS and QNX drives, you may also specify the access privileges to the DOS drive (Read, Write, Modify) for non-super users. Super users always have full access to DOS disks. At the time of this printing, DFS does not support partitions greater than 32Meg in size. DFS does support standard DOS hard disk parti-

tions as well as 360K, 1.2M, 720K and 1.44M DOS formatted floppy disks.

File-type recognition rules.

The rules for recognizing file types have been discussed in the previous section. You may specify the rules to be used when starting DFS, and you may modify them at any time using the DFS command interpreter.

File extension lists.

You may provide lists of file extensions which are to be treated as binary or text files. These lists may be specified on the command line, or they may be contained in files. You may specify as many lists as you wish for both text and binary files.

Non-standard drive characteristics.

Some disk drives do not fully follow the DOS conventions for file structures. It is possible to specify a number of items which will fully describe the disk format to the DOS file system, and thus allow these non-standard drives to be handled by the DOS file system.

1.3.2 Installation

The distribution disk contains the following files:

<code>/cmds/dfs</code>	- The command interpreter
<code>/cmds/dosfsys</code>	- The DOS file system administrator
<code>/config/dfs</code>	- A sample file extension table

Copy these files to your hard disk as follows:

<code>cp s:/cmds/dfs</code>	<code>t:/cmds</code>
<code>cp s:/cmds/dosfsys</code>	<code>t:/cmds</code>
<code>cp s:/config/dfs</code>	<code>t:/config</code>

S is the drive number for the distribution diskette and t is the drive number of the target drive which contains your /cmds and /config directories.

1.3.3 Backing up DOS partitions with DFS

DFS can be used in conjunction with the QNX archiving commands **FBACKUP**, **TBACKUP** and **BACKUP** to archive the information on your DOS partitions. You will need to reference the documentation pertaining to each of these commands for their particular syntax before attempting to save your data. The impor-

tant thing to remember is to make sure that DFS is started with the **+binary** option so that no data translation will occur. Otherwise the the contents of some files may be translated on your archive (because of the translation of CR/LF to RS). The **+binary** option must also be specified when restoring to a DOS partition.

1.3.4 DOS-QNX and QNX-DOS character mapping

The **+rundos** option will force DFS into binary mode and perform character mapping. This mapping is necessary because not all characters in DOS filenames are supported under QNX. Here is a table of the character mappings which will be performed when this option is enabled.

DOS char QNX char

!	A
#	B
\$	C
%	D
&	E
\	F
(G
)	H
-	I
@	J
\\	/
^	K
'	L
{	M
}	N
~	O

1.4 Command Syntax

Syntax:

```
dfs start [node] drive_spec ... [options] [mode_spec] [extension_spec]
dfs add  [node] [drive_spec ...] [extension_spec]
dfs rem  [node] [dos_drive ...] [extension_spec]
dfs mode [node] mode_spec
dfs query [node] [+full_display]
dfs stop [node]
```

drive_spec: dos_drive=qnx_drive[(rwm)][,special_spec]
dos_drive=pseudo_disk_file_name[(rwm)]

options: F=number_files | N=number_buffers
C=cache_timeout | +rundos
+verbose | +Rambo

mode_spec: +text | +bin | +default | D=blt

extension_spec: B=ext,ext... | T=ext,ext... | E=fname

special_spec: cluster_size,num_fat,num_dir_fat_size,fat_offset

The **start** command will start a new copy of the DOS file system administrator. Only one copy may be active on a given node.

The **add** command allows you to add drives or file extensions to an active copy of the DOS file system.

The **rem** command allows you to remove drives or file extensions from an active copy of the DOS file system.

The **mode** command allows you to change the rules being used to recognize DOS text files by an active copy of the DOS file system.

The **query** command allows you to view the full operating environment of an active copy of the DOS file system.

The **stop** command allows you to terminate an active copy of the DOS file system. When this command is issued, no new open requests will be honoured and the administrator will automatically shut down once all active files have been closed.

Options:

- node - Specifies the node where the copy of the DOS file system is active. If this is not specified, the node where the DFS command was issued will be assumed.
- drive_spec - Defines the characteristics of each drive managed by the DOS file system.
- dos_drive - The DOS drive number (a-f).
- qnx_drive - The QNX drive number of the DOS drive (1-8). This may include a node number. It may also be the name of a file which is a DOS pseudo disk. This is explained in the QDOS manual.
- (rwm) - The access mode to the DOS drive for non-super users.
 - r specifies read access to the drive for non-super users.
 - w specifies write access to the drive for non-super users.
 - m specifies modify access to the drive for non-super users.

These permissions apply to all files on the DOS drive. They may be specified in any order. If no permissions are specified, the DOS file system assumes read/write access to the drive.

- special_spec - This allows the specification of information which fully describes the format of a non-standard DOS drive. If this specification is required, ALL items of information must be supplied. See the PC-DOS or MS-DOS Technical Reference Guide for detailed information concerning these items.

<i>cluster_size</i>	Number of sectors in a cluster.
<i>num_fat</i>	Number of File Allocation Tables on the disk.
<i>num_dir</i>	Number of directory entries in root directory.
<i>fat_size</i>	Number of sectors in the FAT.
<i>fat_offset</i>	Sector number of the first sector of the FAT.
- F=*num_files* - Specifies the number of simultaneously open files to be supported. The default and minimum is 10 files.
- N=*num_bufs* - Specifies the number of buffers to be used to hold FAT blocks, directory blocks and cached data. The default and minimum is 20 buffers.

- C=cache_timeout** - This is the idle time allowed on a floppy (removeable media) before DFS will purge its internal cache. The default time is 1 second. This will solve problems related to ill behaved DOS applications which will leave files open even after a media switch.
- +rundos** - This will force DFS into binary mode and perform QNX-DOS, DOS-QNX character mapping. See section 1.3.4 for details.
- +verbose** - This option will cause DFS to dump error messages when an error on file I/O occurs. Most applications handle errors internally and do not want the DFS error messages displayed. If you wish to allow DFS to display these messages, specify this option on the command line.
- +Rambo** - This option will allow QDOS and DFS to run on the same machine concurrently. Because of the potential problems (due to the nature of DOS) with having QDOS and DFS access the same drives simultaneously, you will be forced to set this option if you wish to allow this mode of operation. The general rule of thumb for QDOS/DFS drive access is: **DO NOT ALLOW READ/WRITE ACCESS WITH DFS AND QDOS EACH ACCESSING THE SAME DRIVES.**
- +text** - Specifies that all files are to be treated as text files. Extensions provided using the options **B=**, **E=** and **T=** are ignored. When used with the **mode** command, it allows you to temporarily suspend the default rules in effect.
- +binary** - Specifies that all files are to be treated as binary files. Extensions provided using the options **B=**, **E=** and **T=** are ignored. When used with the **mode** command it allows you to temporarily suspend the default rules in effect.
- +default** - Specifies that the default rules are to be used to recognize DOS files. The file extension is checked in the provided extension lists and if a match occurs then the file is typed as indicated. Otherwise, the type specified in the **D=** option is used. This is the default. This option is typically used to turn off the effect of a **+binary** or **+text** option.

- +full_display - This option is only valid with the **dfs query** command. This will cause DFS to display detailed parameter information pertaining to each drive.
- D=b|t** - Specifies that file extensions which are not contained in the extension lists are to be treated as text **t** or binary **b** files. The default is to assume binary files where the file extension is not recognized.
- B=ext,ext*** - Specifies a list of DOS file extensions to be treated as binary files.
- T=ext,ext*** - Specifies a list of DOS file extensions to be treated as text files.
- E=fname** - Specifies a file containing lists of binary and text file extensions. The file has the following format:

```

" Lines that start with a double quote are comments.
" The keyword text indicates that the following
" extensions refer to text files.
text
tab  asm
tab  bat
tab  lst
tab  txt
" The keyword binary indicates the the following
" extensions refer to binary files.
binary
tab  com
tab  exe
tab
tab  obj

```

Each extension is listed on a separate line and must start with a single TAB character. The blank or omitted file extension may be entered as a TAB by itself.

You may put as many sets of binary and text extensions as you wish in the file. The file **/config/dfs** contains a sample file extension file which you may wish to modify to suit your own needs.

1.5 Sample DFS Configurations

1.5.1 Example I

Assume an environment with 2 floppy disks and a hard disk which has been partitioned into a DOS and QNX partition using the FDISK utility. The two disk partitions have been mounted with the following commands:

```
mount disk 3 /drivers/disk.at pa=qnx
mount disk 4 d=3 pa=dos
```

You would like to start up a DOS file system which will allow you to access DOS files on the two floppy disk drives and on the DOS partition of the hard disk. You want to use the default rules and will use the list of file extensions found in the file `/config/dfs` to identify text files. The required command will be.

```
dfs start a=1 b=2 c=4 E=/config/dfs
```

To list what is on your DOS drive C, you could use the QNX `ls` command as follows:

```
ls c/
```

Once DFS is started, you may access your DOS partitions just as you would your QNX partitions.

If you wish non-super users to have modify permission on the drives you would specify it as follows.

```
dfs start a=1(rwm) b=2(rwm) c=4(rwm) E=/config/dfs
```

You need modify permission to use the `CHATTR` command and to create and delete directories.

1.5.2 Example II

Assume an environment with one floppy disk drive and a DOS pseudo-disk which was created for use by the DOS BIOS emulation commands. In this case, you want to have full access to the floppy disk, but you want to restrict non-super users to read-only access on the pseudo-disk. In addition you want all files to be treated as binary files. The required command will be.

```
dfs start a=1(rwm) c=/dos/pseudo_disk(r) +b E=/config/dfs
```

Later, you decide to add a drive on a remote node to an active copy of a DOS file system while it is running. In this case, you want the default (read-write) access to the drive.

```
dfs add d=[3]4
```

Let's say you then wish to change the meaning of two file extensions from text files to binary files. (One of the extensions to be changed is the blank or omitted extension, which is represented by a comma with no text in front of it) The **mode** command ensures that you are running default rules and file extension checking, not **+b** or **+t**.

```
dfs mode +d  
dfs rem T=,lst
```

OR

```
dfs mode +d  
dfs rem T=  
dfs rem T=lst  
dfs add B=  
dfs add B=lst
```