

Quikcopy

Version 3.41

by

Roland Skinner

Copyright © 1992, 1993, 1994



TABLE OF CONTENTS

1. INTRODUCTION	4
1.1 What is QUIKCOPY?	4
1.2 What's New in QUIKCOPY?	6
1.3 How to Use This Manual	7
2. REQUIREMENTS AND INSTALLATION	9
2.1 Environment Settings	9
2.1.1 RJSTEMP, TEMP and TMP Settings	9
2.1.2 COPYCMD Setting	11
3. FILE COPYING.....	12
3.1 QUIKCOPY with No Parameters	12
3.2 File Copying Parameters	12
3.2.1 Copy Subdirectories	13
3.2.2 Moving Files	15
3.2.3 Copy All Files	15
3.2.4 Prompt Before Copying	16
3.2.5 Overwrite Checking	16
3.2.6 Slowing Down QUIKCOPY	17
3.2.7 Maximal Diskette Filling.....	17
3.2.7.1 <i>Wipe Diskettes</i>	18
3.2.7.2 <i>Split Large Files into Cut-Files</i>	18
3.2.7.3 <i>QC_PASTE</i>	19
3.2.7.4 <i>Maximal Diskette Filling and Subdirectories</i>	20
3.2.8 Copy Batch of Diskettes.....	21
3.2.9 Ignoring Cut-Files	21
3.3 Shelling to DOS	21
3.4 Cancelling Copy Process	22
4. DISKETTE COPYING.....	23
4.1 BIOS vs Direct Access Copying	23
4.2 Copy Multiple Diskettes	24
4.3 Batch Copying of Diskette.....	24
4.4 Force Destination Formatting	25
4.5 Verifying Diskette Data.....	25
4.6 Use of Disk Memory During Copying	25
4.7 Ignore Copy Errors.....	26
4.8 Cancelling Copy Process	26
5. TRULY WILD WILD-CARDS	27
5.1 Truly Wild Wild-Cards Explained.....	27
5.2 Excluding Files	28
5.3 Multiple Source Files	30
5.4 Extended Parent Subdirectory Names.....	30
6. QUIKCOPY PARAMETER LISTING.....	32
7. TIPS AND TROUBLE SHOOTING	35

8. ACKNOWLEDGEMENTS 38

9. REGISTRATION AND CORRESPONDENCE..... 39

 9.1 QUIKCOPY is Shareware..... 39

 9.2 DISCLAIMER 39

 9.3 Contacting RJS Software or the Author 40

1. INTRODUCTION

Thank you for using QUIKCOPY.

This is the twenty-first official public release of QUIKCOPY. It has come a long way since its initial public release in late 1992.

QUIKCOPY started off in the late 1980s as a personal tool to recover files from both damaged diskettes and diskettes that were in bad shape. Back then, it was used on an XT, and did a remarkably good job of copying files that DOS was unable to. In early 1992, it was decided to rewrite QUIKCOPY, and take advantage of the features of the Intel 80386 and i486 architectures. QUIKCOPY made use of extended and expanded memory, via the XMS and EMS specifications, and it soon became apparent that it copied one's files more quickly than DOS's utilities.

It was then that friends saw this little utility. Giving out copies of what was then v1.00 and v1.01, I soon got feedback that QUIKCOPY was a worthwhile and heavily used utility. It was then realised that many other users – not only a circle of friends – could benefit from its use. A much revamped v2.00 of QUIKCOPY was written, and this was released to the world of bulletin boards. Users from all over started asking for new features, multitasker-support and supplying the all-important bug-reports. Within a year, over fifteen releases of QUIKCOPY were released, and QUIKCOPY had progressed from a personal system aid to a “serious” utility.

This release, v3.41, corrects certain bugs that existed in v3.40 of QUIKCOPY. The previous release, v3.40, was the result of a six month break from the regular new-release pattern that QUIKCOPY users became accustomed to. During this period, many new features were recommended by users, with each and every one being noted. Not all recommendations have been included in this version of QUIKCOPY; a twelve month break would have been required for that. QUIKCOPY is now into its third year as a public utility, and versions 3.4x are the culmination of many hours of work, and it is hoped that you, the user, will appreciate the features.

Currently, QUIKCOPY performs both file and diskette copying. In *file copying mode*, QUIKCOPY can copy or move files, include subdirectories, copy any files, maximally fill diskettes, split large files across multiple diskettes and rejoin them later. In *diskette copying mode*, diskettes may be copied in a single pass, to diskette image-files and vice-versa, with diskette access either being direct or through BIOS. These are but a few of QUIKCOPY's capabilities; rest-assured, there is a feature in QUIKCOPY that you'll find useful.

1.1 What is QUIKCOPY?

QUIKCOPY is a quick file and diskette copier, with numerous features. It is a utility which copies one or more files, from one or more subdirectories on one or more drives, to another drive and/or subdirectory. Also, one or more diskettes may be copied to one or more other diskettes, utilizing diskette image-files if necessary.

QUIKCOPY will not give up on read-errors, and will attempt to read a damaged file/diskette until it succeeds. No more Abort, Ignore, Retry, Fail messages. When copying diskettes, one may specify that QUIKCOPY automatically ignore errors after a number of failures.

QUIKCOPY utilizes all available memory in a system – conventional, EMS and XMS are supported – and uses memory extensively to improve copy-performance. When copying diskettes, QUIKCOPY may even utilize hard-disk or ram-disk space to aid in the copying process.

When copying files, one has the ability to move those files instead. Also, files may be specified from a number of sources, as well as be excluded from a general list of files. QUIKCOPY is one of the only utilities that allows one to lock the destination directory. This is useful when you want to copy or move all files of a certain type, e.g. * .BAK, to a specific directory from where they can be used, deleted, etc.

QUIKCOPY allows one to copy or move the contents of one or more subdirectories to a number of diskettes, in such a way as to maximally fill them. Note that as of v3.40, subdirectories within subdirectories are supported. During this process, one may optionally split large files – too big to fit on a single diskette – across multiple diskettes. The reverse process is also possible; files may be copied or moved from a number of diskettes.

Overwrite checking is supported. If you're prone to overwriting important files, this may prove useful. Similarly, one may specify that QUIKCOPY prompt the user before each file, as to whether or not that file should be copied.

During file copying, a greatly improved and extended wild-card system is supported. This system is backward compatible with that of DOS, and is a superset of that provided by 4DOS and NDOS.

When copying diskettes, QUIKCOPY allows one to copy a single diskette to more than one diskette, without having to re-read the source diskette each time. QUIKCOPY allows access to the diskette drives through two methods: through BIOS and through direct access to the drives. The latter method is somewhat faster, but requires fully compatible systems. Diskette image-files – files that essentially contain the entire contents of a diskette – are supported (both as source and target “diskettes”).

An important feature of QUIKCOPY is that it can copy just about any diskette format. QUIKCOPY does not however try to copy diskettes that use special copy-protection formats.

QUIKCOPY has built-in support for various multitaskers, especially DESQview and Microsoft Windows. It regularly releases time-slices during its execution.

QUIKCOPY supports a couple of disk-caches. QUIKCOPY currently support SMARTDRIVE (v4.00+), PC-CACHE (v8.0+) and HYPERDISK. Disk-caches compatible with these caches will also be supported.

1.2 What's New in QUIKCOPY?

Version 3.40 introduced many new features into QUIKCOPY. Most improvements were made to the file copying component of QUIKCOPY. The following is an outline of these new features, along with a brief description of each:

- Performance of copying and moving of files has been greatly improved since the last version. Speed gains of as much as 300% can be gained, especially on systems that do not run disk-caches. A related topic, QUIKCOPY now chooses to use XMS memory before EMS memory.
- Diskette copying has also seen improvements in speed. As well as improving the speed of diskette copying, a new approach to copying has been included. Now, QUIKCOPY will, by default, directly access the diskette drives of a system. This, however, does require that the system is 100% compatible with IBM's PC series. One can, if the need arises, specify that QUIKCOPY revert to going through BIOS while copying diskettes.
- Maximal diskette filling has seen many improvements. The general diskette filling algorithm has been improved and is somewhat faster.
- Subdirectories are now supported when maximally filling diskettes. Subdirectories within subdirectories can now be copied onto diskettes, and QUIKCOPY will create the necessary directory paths.
- QUIKCOPY will now split large files, too big to fit on a single diskette, across multiple diskettes. File splitting is optional, and has to be enabled in order to utilize this feature. Optionally, QUIKCOPY can copy a small program, QC_PASTE.EXE, onto the diskette that contains the first *cut-file* (see later) of the files that make up the original file. This program is capable of rejoining the cut-files together again. QUIKCOPY is capable of joining these files together, by default. (This may be disabled.)
- When filling diskettes, one may specify that each diskette inserted to be filled, may be erased before the filling process begins.
- QUIKCOPY now performs the reciprocal function of diskette filling. It can copy the contents of a number of diskettes.
- During diskette filling, while joining cut-files or copying from multiple diskettes, when QUIKCOPY prompts for the next disk, one may temporarily shell to DOS.
- Prompting before copying or moving each file has now been added. QUIKCOPY will prompt the user before each file, after which a simple "yes" or "no" answer is required.
- QUIKCOPY now performs overwrite checking. If enabled and a file being copied already exists on the destination, then the user will be prompted whether or not to overwrite the file. By default, overwrite checking is disabled; this provides backward compatibility with previous versions of QUIKCOPY.

In keeping with DOS 6.20 conventions, QUIKCOPY supports the COPYCMD environment setting.

- A far more improved wild-card system has been incorporated into QUIKCOPY. QUIKCOPY has “truly wild wild-cards”. This wild-card system is backward compatible with that of DOS, and is a super-set of that provided with 4DOS and NDOS. File-exclusions are supported, and multiple source parameters may be specified.
- The command-line parser has been improved. No longer are spaces required between options passed to QUIKCOPY (except, of course, between source and destination parameters).
- Beeps and bells have been added during diskette copying and while filling diskettes.
- Numerous problems inherent in previous versions of QUIKCOPY have been removed. Also, screen output has been tuned to produce less flicker – especially under multitasking environments, such as in a Windows DOS-box – and screen output is also somewhat faster. Extra long filename paths will no longer make QUIKCOPY’s output look quite unsightly.
- QUIKCOPY’s diskette-filling and multiple diskette batch-copying facilities may be used to perform BACKUP and RESTORE functions.
- Finally, as users familiar with QUIKCOPY have probably noticed, the documentation has been totally rewritten.

This release, v3.41, corrects a number of bugs that remained or were introduced in v3.40. Also, high-quality documentation has been added to QUIKCOPY; this is available separately. (See the file README.TXT for more information.)

1.3 How to Use This Manual

This manual is broken up into a number of sections.

Section 1, this section, gives a brief introduction to QUIKCOPY and outlines its various capabilities. A list of new features in the current release is also given.

Section 2 is a short section describing the requirements of QUIKCOPY, and gives a few recommendations on how it should be installed.

Section 3 is a rather lengthy section. It describes the various file copying features, and how they can (and at times, should not) be used. Various examples are included.

Section 4 describes the disk copying features of QUIKCOPY.

Section 5 gives a detailed description on how the wild-card system works. Many examples are included to clarify its various benefits.

Section 6 contains a listing, in alphabetical order, of the various options available with QUIKCOPY.

Section 7 contains a short section on common problem shooting questions, and how to eliminate them.

Section 8 acknowledges other software/hardware mentioned in this documentation.

Finally, **Section 9** contains information on how to register QUIKCOPY, contact the author or *RJS Software*, and other related information.

At times, the left margin of this documentation contains **NB**. This indicates that an important warning, related to some aspect of QUIKCOPY is given in the text alongside.

It has been endeavoured to keep a consistent style in this high-quality documentation. *Italics* and **bold-face** are used to highlight section-names, key-words, concepts, etc. Filenames, commands as typed at the command-prompt, and text that may be displayed on the screen are presented in a type-written font, e.g. `FILENAME.EXT`. Program names and/or versions are usually printed in a Helvetica font, e.g. QUIKCOPY.

2. REQUIREMENTS AND INSTALLATION

QUIKCOPY will work on any PC containing one of the Intel 80x86 family of processors (or compatibles), ranging from a bare-boned XT to a power-house Pentium.

A realistic minimum amount of memory for QUIKCOPY is 256K, however 640K is recommended. The more memory available in the system, the more beneficial QUIKCOPY will be. QUIKCOPY will access additional memory via the XMS and EMS memory specifications.

The system must be running DOS 3.20 or higher. QUIKCOPY has been tested with DR DOS 6.0, and was found to work successfully.

If an EMS memory manager is operational in the system, it must support the EMS v4.00 (or above) specification.

It is recommended that QUIKCOPY be installed in a directory that is included in your computer's PATH setting. Alternatively, one may set one or more aliases to QUIKCOPY and/or its various options. (Some users set an alias DISKCOPY for QUIKCOPY /D.)

2.1 Environment Settings

A number of environment settings may be used to improve the execution of QUIKCOPY, or eliminate problems that may arise while executing QUIKCOPY.

2.1.1 RJSTEMP, TEMP and TMP Settings

Probably the most important environment setting used during QUIKCOPY's execution is that specified in either of the RJSTEMP, TEMP or TMP environment settings. All temporary files are created and stored in the directory specified by one of these settings.

QUIKCOPY will first try to use the directory specified by the RJSTEMP environment variable. If that fails, the directory pointed to by TEMP will be used. Otherwise, the directory pointed to by TMP will be used. If neither of these three environment variables exists in the environment, then QUIKCOPY will try to create these temporary files in the current directory.

It is recommended that you *do* add one of these settings to your environment. Often, these environment variables are already present in one's environment block, and no modifications need to be made. For example, Windows and many other programs create such an environment setting, by default, during their installation. Check whether your system includes such a setting in your AUTOEXEC.BAT file. If one exists, you need not worry about it anymore (unless of course QUIKCOPY complains).

To install one of these environment variables, simply add a line similar to the following in your AUTOEXEC.BAT file:

```
SET RJSTEMP=C:\TEMP
```

In this case, the RJSTEMP environment variable is set in the AUTOEXEC.BAT. The temporary directory is C:\TEMP. Note that there is no need to add a trailing backslash (“\”). In general these environment variables are set as follows:

```
SET RJSTEMP=d:path
```

or

```
SET TEMP=d:path
```

or

```
SET TMP=d:path
```

where d:path is the drive and directory path of the temporary directory.

It is recommended that you do not set one of the temporary directory environment variables to refer to a diskette drive, or other form of removable medium. Also, if it can be avoided, try not to use a remote drive (for example, in a networked environment). If you have sufficient memory, and a large enough ram-drive, the temporary directory may be set to that ram-drive. Here, sufficient memory is somewhere in the range of 1.5M to 2M of free space (enough to store a diskette image).

Be sure that the temporary directory specified in either of these environment variables exists.

One use of a temporary file is during diskette copying. If there is insufficient memory to store the entire contents of a diskette, QUIKCOPY will try to store the remaining contents of the diskette in a temporary data file.

NB One important reason to install one of these temporary variables in your environment is that QUIKCOPY includes a file manager that allows it to effectively open more files than allowable by the system on which it is running. It creates a temporary file as a sort of insurance policy against it ever being unable to open/create a file. If QUIKCOPY is unable to create such a temporary file, it will unceremoniously halt displaying the error-message:

```
ERROR - Could not install file manager!
```

This error commonly arises when there is no such temporary variable in the environment, and the current directory belongs to a *read-only* drive, e.g. CD-rom, write-protected diskette or read-only access network drive. Another cause of this problem is if the directory path specified in the particular environment variable does not exist.

2.1.2 COPYCMD Setting

The COPYCMD environment variable is used to set one's overwrite action preferences, when copying or moving files. This variable was introduced with DOS 6.20, and QUIKCOPY follows its specification.

There are two settings for this variable:

- /Y – Always overwrite file, if it already exists on the destination.
- /-Y – If a file already exists on the destination, prompt user whether or not file may be overwritten.

To set this environment variable, simply add the lines:

```
SET COPYCMD=/Y
```

or

```
SET COPYCMD=-Y
```

to your AUTOEXEC.BAT file. Because QUIKCOPY overwrites files by default, it is not necessary to include /Y setting. However, if using DOS 6.20, you may already have it set for use with DOS's file copying and moving commands.

For more information about various overwrite options, see **Section 3.2.5**.

3. FILE COPYING

QUIKCOPY contains a number of file copying options and features. By default, QUIKCOPY goes into file copying mode.

3.1 QUIKCOPY with No Parameters

If no parameters are given, a short description of the various command-line parameters is given. It looks something like the following:

```
QUIKCOPY [[d:] [path] source [[d:] [path] destination]
          [[</S | /E> [/L]] [/F [/C [/IP]] [/W]] [/B] [/M] [/A] [/NJ]
           [/P<Y|N>]] [/DELAY[:ms]] [</O[<Y|N>[A]] | /Y | /-Y>]
          ] |
          [/D [/BIOS] [/M<+|->] [/B] [/F] [/V<F|W|A|N>] [/N]
           [/I[:num]]
          ]
          ] | [/HELP | /?]
```

This is essentially a summary of all command-line parameters available. Options in square brackets, such as [/L], are optional. Options between “<” and “>”, such as <+|-> in /M<+|->, are required. Any two or more options separated by a vertical bar (“|”) require that one of the options be used. For example, with </S | /E>, either /S or /E may be used (never both). These “rules” may be applied recursively, as indicated in the parameter summary above.

3.2 File Copying Parameters

If any of the first group of parameters (as shown in the summary above) exist, then QUIKCOPY will copy (or move) a number of files. In all cases, the source-parameter must be included. The source parameter may be any combination of source specifications allowed by the wild-card system (see **Section 5**). If a source parameter is not specified, QUIKCOPY will terminate displaying an appropriate error.

Optionally, one may specify a destination to which files are to be copied. If no destination is specified, then QUIKCOPY will assume that the current directory is the destination for any files to be copied or moved.

NB The destination parameter does not follow the wild-card convention described in **Section 5**. Standard DOS wild-cards are, however, allowed. QUIKCOPY will insert any necessary characters where wild-card characters, “*” or “?”, appear in the destination parameter. Note that if no character can be substituted for a particular wild-card character, QUIKCOPY will replace that character with an underscore (“_”).

All other file copying parameters may be added optionally. The following subsections describe the use of these parameters.

3.2.1 Copy Subdirectories

QUIKCOPY offers three command-line parameters, that are related to copying subdirectories and their files. Specifying either /S or /E will indicate that QUIKCOPY should copy (or move) files included in the subdirectories within the source subdirectory (or subdirectories). If either of the two parameters (/S or /E) is used, the other parameter cannot be used.

Specifying /S on the command-line indicates that QUIKCOPY should copy subdirectories, but only those that contain files to be copied (or moved). /E on the other hand, specifies that QUIKCOPY must copy all subdirectories, regardless of whether or not there are any files in these subdirectories to be copied.

For example, suppose you have the following directory structure:

```
C:\MYFILES
├── DOCUMENT
│   ├── TXTFILES
│   │   ├── file0001.txt
│   │   ├── file0002.txt
│   │   └── file0001.tex
│   └── DOCFILES
│       ├── file0001.doc
│       └── file0002.doc
├── BAKFILES
└── EXEFILES
    ├── file0001.exe
    └── file0002.exe
```

and execute the following command:

```
QUIKCOPY C:\MYFILES /S
```

Suppose that the destination (the current directory in this case) is D:\HISFILES. Suppose also that the destination was empty. After copying, the destination will have the following structure:

```
D:\HISFILES
├── DOCUMENT
│   ├── TXTFILES
│   │   ├── file0001.txt
│   │   ├── file0002.txt
│   │   └── file0001.tex
│   └── DOCFILES
│       ├── file0001.doc
│       └── file0002.doc
└── EXEFILES
    ├── file0001.exe
    └── file0002.exe
```

Notice that the empty subdirectory C:\MYFILES\DOCUMENT\BAKFILES was not copied.

Now, suppose we had typed:

```
QUIKCOPY C:\MYFILES D:\HERFILES /E
```

Again, suppose that the destination was empty. After copying, the destination will have the following structure:

```

D:\HERFILES
├── DOCUMENT
│   ├── TXTFILES
│   │   ├── file0001.txt
│   │   ├── file0002.txt
│   │   └── file0001.tex
│   ├── DOCFILES
│   │   ├── file0001.doc
│   │   └── file0002.doc
│   └── BAKFILES
└── EXEFILES
    ├── file0001.exe
    └── file0002.exe

```

Notice that this time the empty directory was also copied.

When copying subdirectories, you may wish to copy the files from a number of subdirectories, but store them in a single subdirectory. To do this, you specify either of the include-subdirectory parameters (/S or /E), and add the /L command-line parameter. This parameter indicates that you are locking the destination subdirectory. All files will be copied to the locked subdirectory.

For example, suppose again that we have the directory structure C:\MYFILES as shown above. This time we wish to copy all document files (C:\MYFILES\DOCUMENT), and copy them to a new directory C:\MYFILES\MYDOCS. Also, suppose that the current subdirectory is C:\MYFILES.

We execute the following command:

```
QUIKCOPY DOCUMENT\*.TXT;DOCUMENT\*.DOC MYDOCS /S/L
```

After QUIKCOPY completes execution, the directory structures will look something like:

```

C:\MYFILES
├── DOCUMENT
│   ├── TXTFILES
│   │   ├── file0001.txt
│   │   ├── file0002.txt
│   │   └── file0001.tex
│   ├── DOCFILES
│   │   ├── file0001.doc
│   │   └── file0002.doc
│   └── BAKFILES
└── EXEFILES
    ├── file0001.exe
    └── file0002.exe
MYDOCS
├── file0001.txt
├── file0002.txt
├── file0001.doc
└── file0002.doc

```

Now, a few points have to be noted here. QUIKCOPY will create a subdirectory if it doesn't exist, and expects more than one file to be copied. In this case, it created the subdirectory C:\MYFILES\MYDOCS. Notice also that we specified two source parameters. Multiple source parameters are separated by a semi-colon (";"); no spaces are allowed. All the *.TXT and *.DOC files were copied to C:\MYFILES\MYDOCS.

3.2.2 Moving Files

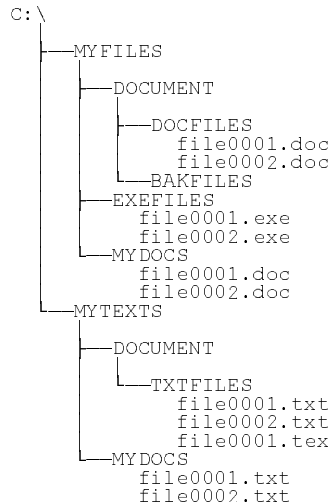
QUIKCOPY has the ability to move files, instead of copying them. By adding the /M parameter, QUIKCOPY will move files from the source to the destination. When copying subdirectories, if QUIKCOPY moves all the files (and subdirectories) within a subdirectory, that subdirectory will be deleted from the source.

If the files are being moved from one subdirectory on a drive to another directory on the same drive, then QUIKCOPY will do a fast move, in which it doesn't actually read the contents of the files, but simply moves them. When moving to a different drive, the files are first copied to the destination, then deleted from the source.

For example, suppose that our directory structure is that which resulted from the locked subdirectory copy above. Suppose that we are in the root directory of drive C: . This time we execute:

```
QUIKCOPY MYFILES\*.T* MYTEXTS /S/M
```

After execution, the directories will have the following structure:



As can be seen, QUIKCOPY created the necessary subdirectories C:\MYTEXTS, C:\MYTEXTS\MYDOCS and C:\MYTEXTS\DOCUMENT\TXTFILES. The *.TXT and *.TEX files were moved to the newly created subdirectories. Also, QUIKCOPY deleted the original subdirectory C:\MYFILES\DOCUMENT\TXTFILES.

3.2.3 Copy All Files

By default, QUIKCOPY will not copy files or subdirectories that have their *hidden* and *system* attributes set. In order to copy all files and subdirectories, including hidden and system, simply add the /A parameter.

For example, suppose there was a hidden subdirectory C:\MYFILES\DOCUMENT\SECRETS that contained a number of confidential documents. The actual directory structure already presented in **Section 3.2.1** is as follows:

```

C:\MYFILES
├── DOCUMENT
│   ├── TXTFILES
│   │   ├── file0001.txt
│   │   ├── file0002.txt
│   │   └── file0001.tex
│   ├── DOCFILES
│   │   ├── file0001.doc
│   │   └── file0002.doc
│   ├── SECRETS
│   │   ├── secret01.doc
│   │   └── secret02.doc
│   └── BAKFILES
└── EXEFILES
    ├── file0001.exe
    └── file0002.exe

```

This time, we want to copy all (and we mean all) *.DOC files into the D:\HISFILES subdirectory. We assume this subdirectory is empty. The following command is executed:

```
QUIKCOPY C:\MYFILES\*.DOC D:\HISFILES /S/L/A
```

After execution, the destination directory structure is as follows:

```

D:\HISFILES
file0001.doc
file0002.doc
secret01.doc
secret02.doc

```

3.2.4 Prompt Before Copying

Often, one cannot quite remember the name of the file that you wish to copy. At the same time you wish to copy a number of other files; these files' names too, you cannot quite remember. This is where QUIKCOPY's "prompt-before-you-copy" parameter can help. By specifying /P, /PY or /PN you specify that QUIKCOPY should ask you whether to copy a file or not.

Specifying either /P or /PY will allow these prompts, along with a default answer of "yes". /PN allows these prompts, but with a default answer of "no".

NB Note that only one of these parameters may be chosen at a time.

Choosing which of these parameters to use, is up to the user's discretion. Probably, a good approach would be to use /P or /PY when you wish to copy most of the files, whereas /PN would be used when only a few of the files are to be copied.

3.2.5 Overwrite Checking

QUIKCOPY has four methods with which to handle overwrite checking:

- Always overwrite a file. This is the default approach.
- Never overwrite a file.
- Check with user before overwriting file. Default answer to question will be "yes".
- Check with user before overwriting file. Default answer will be "no".

Specifying `/OYA` or `/Y` will cause **QUIKCOPY** to always overwrite a file if it already exists on the destination. Note, that this is the default approach taken, unless of course the `COPYCMD` environment variable is otherwise set (see **Section 2.1.2**).

Specifying `/ONA` informs **QUIKCOPY** never to overwrite a file that already exists on the destination.

Specifying either of `/O`, `/OY` or `/-Y` tells **QUIKCOPY** to check with the user in case a file is to be overwritten. The default answer to the question will be “yes”.

Specifying `/ON` specifies that **QUIKCOPY** should prompt the user for permission to overwrite a file. Default answer will be “no”.

NB Note that the `/Y` and `/-Y` parameters are added for compatibility with DOS 6.20.

NB Only one of these parameters may be specified at a time.

NB Note that if a file on the destination is read-only, **QUIKCOPY** will not overwrite it.

3.2.6 Slowing Down QUIKCOPY

Often, users of **QUIKCOPY** “suffer” from the “hand is quicker than the eye” syndrome. As requested by some users, people want to know what is going on, and if **QUIKCOPY** is rapidly copying files, the screen will scroll so fast that only the last few files copied or moved will be seen.

If one wants to “control the hand”, the `/DELAY` or `/DELAY:ms` parameters may be used. `/DELAY` specifies that a delay of 500ms should occur after copying (or moving) each file. To specify a custom (preferred) delay period, `/DELAY:ms` can be specified, where `ms` is in the range of 1 to 1000.

3.2.7 Maximal Diskette Filling

One of **QUIKCOPY**’s most attractive features is its ability to copy or move the contents of one or more subdirectories to a number of diskettes, leaving as little space as possible on the diskettes. With the release of v3.40, **QUIKCOPY** supports filling these diskettes with subdirectories and their files.

To enable maximal diskette filling, all one need do is add the `/F` parameter to the command-line. The destination must be a diskette drive. **QUIKCOPY** will terminate with an appropriate error if the destination is not a diskette drive.

QUIKCOPY will then proceed with copying or moving files to the diskettes. When a diskette is full, **QUIKCOPY** will prompt the user to insert another diskette to be filled. At this prompt, one may optionally shell to DOS (see **Section 3.3**). When the next diskette is ready to be filled, simply press any key other than `Alt D`, `Esc`, or one of the Break keys, `Ctrl C`, `Ctrl Pause` or `Ctrl Break`, and **QUIKCOPY** will proceed to fill the next diskette. This process continues until all files have been copied (or moved) or the user cancels the process.

3.2.7.1 *Wipe Diskettes*

Often, when filling diskettes, one may run out of blank diskettes. Usually, a person has a number of diskettes lying around that contain useless information. It would then be desirable to use these diskettes by erasing their contents and filling them with other files.

Sure, you can shell out to DOS, erase the diskette, return to QUIKCOPY and resume the diskette filling process. This is quite time-consuming, and thus, QUIKCOPY allows one to specify that it wipes the contents of diskettes before filling them. By adding `/W` to the command-line, QUIKCOPY will automate this process.

NB It is important that you use this parameter with care during the diskette-filling process. If one were to accidentally erase a diskette previously filled, you would have to restart the diskette-filling process. If moving files, this action could result in loss of important data.

NB The `/W` parameter can only be used when maximally filling diskettes.

3.2.7.2 *Split Large Files into Cut-Files*

Often, when maximally filling diskettes, one may have one or more files that are way too large to fit on a single diskette. With versions of QUIKCOPY prior to v3.40, these files simply could not be copied across. QUIKCOPY can now optionally split-up these large files, and write them across multiple diskettes.

QUIKCOPY does this by creating *cut-files*. A cut-file is essentially a portion of one of these large files, and a group of these cut-files is required to restore the original file to its previous state. When splitting a file, QUIKCOPY creates a number of these cut-files, as well as a *stub-file*. This stub-file has the same name as the original file that was split. It is the first file copied to the first diskette that contains a cut-file of the file that was split-up.

NB Do not erase the stub-file. Without it, QUIKCOPY and QC_PASTE (see **Section 3.2.7.3**) cannot restore the original file to its “former glory”.

The cut-files created by QUIKCOPY are of the form `QCOPYCUT.###` where `###` is a number from 0 to 999, padded with zeros. At the beginning of each cut-file, QUIKCOPY stores important information about the cut-file, including the original file’s name. This information is used to ensure that one does not reconstruct previously split files from incorrect cut-files.

NB One cannot restore a previously split file, by simply appending the cut-files together (in order).

To enable file-splitting, add the `/C` parameter to the command-line.

NB The `/C` parameter can only be specified when maximally filling diskettes.

One can use maximal diskette filling, together with this ability to split large files, to copy a single large file to a diskette. Simply give the file's name as the source parameter, and add the /F and /S parameters.

QUIKCOPY can automatically detect files that were previously split. It will automatically reconstruct these files at a later stage. During this reconstruction process, QUIKCOPY also provides the ability to shell to DOS, between swapping diskettes.

NB Note that QUIKCOPY will not erase stub-files and cut-files when reconstructing a previously split file, and files are being moved. This is to ensure that if a problem occurs one or more cut-files later, the initial portions of the file being reconstructed are not lost.

NB When maximally filling diskettes, QUIKCOPY will not reconstruct the various constituent portions of a file. They will be copied (or moved), just as if the /NJ option is in effect (see **Section 3.2.9**).

3.2.7.3 QC_PASTE

Using QUIKCOPY to split large files across multiple diskettes is all “well and fine”, but take a look at the following scenario.

You are in charge of a large database, for which the main database file is never smaller than ten megabytes (10M). Now, your boss has asked that you supply a number of your company's branches, spread across the country, with a copy of the data contained in that database. Now, the various branches all use different brands of PCs, different software tools, and as usual, are running different versions of DOS itself. DOS, true to its form, does not like backups from different versions of DOS, especially if they're different OEM versions; so backing up is out. Well, QUIKCOPY comes to the rescue. Or does it? Do the other branches have a copy of QUIKCOPY? How will they reconstruct the files that were split?

This is where QC_PASTE comes into play. QC_PASTE is a small utility whose soul task is to reconstruct files that were previously split-up by QUIKCOPY. By adding the /IP parameter, QUIKCOPY will copy QC_PASTE.EXE to the diskette containing the stub-file. It will be copied before the stub-file on the first destination diskette containing a cut-file for a particular file that was split. If you're uncertain whether or not the recipient of your data-files has a copy of QUIKCOPY, and you're using the file-splitting capabilities, simply specify that QUIKCOPY copy QC_PASTE.EXE to the destination diskettes containing these files.

QC_PASTE takes either one or two parameters when executed:

```
QC_PASTE [d:][path]source [[d:][path]destination]
```

As with QUIKCOPY, if no destination is given, QC_PASTE assumes the current directory is the destination; also, the reconstructed file will have the same name as the original file. QC_PASTE will not reconstruct a file if it already exists on the

destination. The source parameter is the stub-file's name (same name as the file to be reconstructed).

NB Do not rename a stub-file. `QC_PASTE` (and `QUIKCOPY`) will not be able to reconstruct the file. If you want to reconstruct a file to a file with a different name, specify the new file's name as the destination parameter.

NB Unlike `QUIKCOPY`, `QC_PASTE` does not support wild-cards. Full filename specifications must be given. This also means that only one file may be reconstructed at a time with `QC_PASTE`.

NB The `/IP` parameter may only be used when maximally filling diskettes and when it has been decided to split large files; when `/F` and `/C` are being used.

NB Unlike `QUIKCOPY`, which requires a minimum of DOS v3.20, `QC_PASTE` will work with DOS versions 3.00 and above.

3.2.7.4 *Maximal Diskette Filling and Subdirectories*

A short discussion is required to clarify some aspects of how `QUIKCOPY` maximally fills diskettes with the contents of subdirectories.

At the outset, it is important to note that when subdirectories are involved (`/S` or `/E` specified), `QUIKCOPY` does not perform an optimal fill. In order to ensure that no more time is lost calculating a viable fill, than when it optimally fills diskettes with a single directory's files, heuristics are employed to speed up the process.

The number of combinations to consider when subdirectories are involved would increase the computation time required by a number of orders of magnitude. Thus, the heuristics used by `QUIKCOPY` try to over-estimate the amount of space required, and then fill in what's free thereafter. One will notice that `QUIKCOPY` takes a number of passes reading from the source drive, compared with the single pass when copying a single directory's files.

NB When the diskette is almost full – when only a few clusters are free (usually one or two) – `QUIKCOPY` performs a linear search trying to find a file that it can fit into the remaining free diskette-space. One will often see the following message:

```
Calculating optimal diskette-fill settings...
```

At this time, `QUIKCOPY` may seem to have slowed down, or worse, crashed. This has not happened. You should notice disk access. `QUIKCOPY` is searching for the file that will fill the free space. (See **Section 7** for a tip on how to speed up this process.)





NB Note, the implications discussed in this section do not apply when using a locked destination, i.e. `/L` in conjunction with `/S` and `/E`.

NB When the destination specified is a subdirectory on the diskette, `QUIKCOPY` does take some precautions similar to those applied with subdirectories.

3.2.8 Copy Batch of Diskettes

QUIKCOPY has the ability to perform the opposite process of maximally filling diskettes. It can copy (or move) the files from a number of source diskettes, to a single destination.

By adding the /B parameter, QUIKCOPY will copy all the files that match the source-specification from each diskette. Once a diskette has been copied, QUIKCOPY will prompt the user to insert the next diskette to copy from. As with diskette filling, one may shell to DOS at this stage (see **Section 3.3**).

If all the diskettes that were to be copied (or moved) have been used, by pressing , or one of the Break keys, ,  or , this process may be stopped (cancelled).

Note that this feature need not be used only with diskettes that were previously maximally filled. Any batch of diskettes may be used.

NB The source drive must be a diskette drive when using this parameter. If not, QUIKCOPY will simply ignore this parameter.

3.2.9 Ignoring Cut-Files

At times, one may wish to ignore the fact that the stub-file, cut-files and optional QC_PASTE.EXE file “belong” to a previously split-up file; for whatever reason, one may wish to copy the files themselves.

By adding the /NJ parameter, QUIKCOPY will ignore these files’ special status, and simply copy them.

Note that by default, QUIKCOPY will simply reconstruct files, ignoring QC_PASTE.EXE if it exists.

3.3 Shelling to DOS

As described in some of the previous section, QUIKCOPY provides the ability to shell to DOS. This feature provides the user with the ability to perform maintenance tasks, such as formatting diskettes required for diskette filling, erasing files to free disk-space, and many other tasks.





While in a DOS shell, QUIKCOPY may be re-executed to perform special tasks, etc. QUIKCOPY does modify your command-prompt, so you’ll know that it is still lurking in the background.








QUIKCOPY tries to release as much memory as possible for other utilities executed in the DOS shell. Swapping technology is used to further increase the amount of memory QUIKCOPY frees up for other software. In fact, under favourable circumstances, QUIKCOPY will shrink itself to as little as 688 bytes along with additional bytes occupied by the environment. QUIKCOPY also releases most of the XMS and/or EMS memory it previously had for its use (if any).

NB Be sure not to run TSRs, and other programs that similarly allocate memory, and do not release it after returning to the command-prompt. If you do run such programs, be sure that they return the system to the state it was in when QUIKCOPY shelled to DOS, before returning to QUIKCOPY. If QUIKCOPY cannot successfully restore itself to its previous state, it will unceremoniously terminate itself, without any warning. Worse still, your system may crash under these circumstances.

Remember to type in EXIT to return to QUIKCOPY, after shelling to DOS.

3.4 Cancelling Copy Process

At any stage during the copying process, QUIKCOPY may be terminated by pressing , or one of the Break keys, ,  or . Often, one may be prompted whether or not to continue copying or moving other files. At times, QUIKCOPY will simply terminate displaying a summary of its success.

It may be necessary to press either of these key-combinations if QUIKCOPY is taking too long to copy (or move) a particular (often damaged) file. The Abort, Ignore, Retry, Fail message is not displayed while QUIKCOPY is running. After pressing one of these keys, a message indicating that a certain file was not copied will be displayed, and will ask whether or not to continue copying other files. Pressing  will continue; otherwise, pressing , , , or one of the Break keys, ,  or , will stop the copying process.

Note that this message may also be displayed if an error, such as the destination being full, occurs. A similar response will be required.

4. DISKETTE COPYING

Diskette copying mode is enabled by specifying the /D parameter. QUIKCOPY has a number of diskette copying features. This section and its sub-sections describe these features.

QUIKCOPY is a single pass diskette copier. This eliminates the need to continually swap diskettes, especially in single diskette drive systems. If there is insufficient available memory to store the contents of a particular diskette, QUIKCOPY will create a temporary file that will contain the remainder of the diskette's image (see **Section 2.1.1**). If QUIKCOPY runs out of memory, and cannot successfully utilize disk-space to store the remainder of the diskette's image, it will revert into a multi-pass diskette copier.

QUIKCOPY will not attempt to write to write-protected diskettes. Precautions are taken to ensure that a diskette is not write-protected.

When in diskette copying mode, QUIKCOPY has two different approaches to copying diskettes, viz. through BIOS and by directly accessing the diskette drives. The pros and cons of these methods are discussed in **Section 4.1**.

QUIKCOPY can utilize and create diskette image files. These are essentially files that contain an exact copy of the entire contents of a diskette. This is especially useful when wishing to transfer a diskette across electronic communications media, e.g. a modem and telecommunication lines, to someone situated a great distance away. Sending a diskette via mail, may take much longer than using a simple telephone call.

QUIKCOPY is capable of copying just about any diskette. Non-standard formats such as 1.72M 3.5" diskettes, 800K low-density diskettes, as well as many other formats created by programs such as MAXI Form and FDFormat are supported. QUIKCOPY can also copy 1.44M 3.5" diskettes onto 1.2M 5.25" diskettes, formatting them to 1.44M. QUIKCOPY does perform precautionary checks to ensure that a particular format is "safe" for a particular diskette drive; for example, one cannot possibly expect to create a 1.44M diskette in a 360K drive.

4.1 BIOS vs Direct Access Copying

The two methods of diskette copying at QUIKCOPY's disposal both have their advantages and disadvantages. Essentially, what is involved is a trade-off between speed and compatibility.

Going through BIOS, QUIKCOPY is (virtually) guaranteed to be able to work with a particular system. It does, however, mean that QUIKCOPY is no longer in control of how it accesses the diskettes.

Accessing the diskette drives directly, QUIKCOPY is able to gain speed from eliminating much of the redundant diskette-access performed by BIOS and its various

functions. However, incompatibilities inherent in particular systems may creep in, and as a result, going directly to the hardware will be a futile attempt.

Although the performance of the diskette copying process has been improved in BIOS mode, more gains in speed are to be obtained by going directly to the hardware.

NB It is recommended that one first test whether or not one's system is compatible with QUIKCOPY's direct access method of diskette copying. Most systems will be compatible, but one cannot blindly expect to be a "member of the masses".

By default, QUIKCOPY adopts its direct access mode of copying. Some systems will fail to work in this mode. Common examples, are systems that do not have their diskette controllers installed with the hardware ports set at 03FXh. Also, it is assumed that the diskette controller uses DMA channel #2. QUIKCOPY should not have any problems with different interrupts being used by various diskette controllers.

NB On systems that cannot be used with the direct access approach, the /BIOS parameter should be added to the command-line.

4.2 Copy Multiple Diskettes

By default, QUIKCOPY allows the user to copy multiple diskettes. After copying a diskette, QUIKCOPY will ask the user whether or not to copy another diskette.

The /M+ parameter enables multiple diskette copying.

If one wishes to only copy a single diskette, the /M- parameter may be specified. QUIKCOPY will terminate after copying a single diskette.

When copying to a diskette image file, QUIKCOPY assumes the /M- state; QUIKCOPY will only copy the diskette once.

NB Note that only one of these parameters (/M+ and /M-) may be specified at one time.

4.3 Batch Copying of Diskette

QUIKCOPY has the ability to copy a single diskette to more than one other diskette, without having to reread the source diskette.

By adding the /B parameter, one enables batch-copying. QUIKCOPY will copy a diskette to completion, and ask whether or not to copy the same source diskette to another diskette.

Note that even if multiple diskette copying is disabled, one may still perform batch copying; but only of a single source diskette.

NB If there is insufficient memory and disk-space to store the image of a diskette, batch-copying will require that the source diskette be read multiple times.

4.4 Force Destination Formatting

QUIKCOPY will, by default, inspect the diskette to determine whether or not a diskette format is required. This process is fallable, especially if a diskette was only partially formatted previously.

Also, one may wish to format a diskette to ensure that it is correctly formatted. One may force QUIKCOPY to format destination diskettes, by adding the /F parameter.

4.5 Verifying Diskette Data

QUIKCOPY provides four levels of data verification that may be performed when writing to a destination diskette.

These are:

- Always verify. After both formatting and writing to the diskette, QUIKCOPY will verify that the data on the diskette is in a stable state.
- Verify after formatting. This is the default setting. QUIKCOPY will verify that formats are successful, but will not verify that writes to the diskette succeed.
- Verify after writing. QUIKCOPY will verify data after writing to the diskette, but not after formatting.
- Never verify. No verification, at all, will be performed.

By adding the /VA parameter, QUIKCOPY will always verify. Specifying the /VF parameter indicates that it should verify only after formatting. /VW indicates that verification should only be performed after writing. The /VN parameter may be used to disable all verification.

It is up to the user's discretion which level of data security should be adopted. A speed vs security tradeoff is involved. Maximum security is obtained using the /VA parameter, however diskette copying will be a lengthy process. Fastest copying is obtained using the /VN parameter, however minimal security (if any) is obtained. It is up to the user to strike a balance.

NB Note that only one of these verify parameter may be specified at one time.

4.6 Use of Disk Memory During Copying

As indicated in the introduction of **Section 4**, QUIKCOPY will make use of available disk-space if there is insufficient memory to store the image of a diskette being copied. At times, however, one may not wish to use this disk-space; e.g. perhaps there is insufficient disk-space, you have a diskette only system, or the hard-drive is too slow.

By adding the /N parameter to the command-line, one can effectively disable QUIKCOPY's use of disk-space as virtual memory.

Note that if copying to or from a diskette image file, QUIKCOPY will automatically disable its usage of disk-space.





4.7 Ignore Copy Errors








By default, QUIKCOPY will read the contents of a diskette, retrying on failures until it succeeds. Often, QUIKCOPY is able to recover the contents of the bad sectors on these diskettes. However, some diskettes may be so badly damaged that it cannot possibly recover the data. One may specify that QUIKCOPY ignore errors after a number of failures.

By adding the /I or /I:num parameters, one may specify that QUIKCOPY ignore such errors after a certain number of failures. With the /I option, QUIKCOPY will ignore an error if it fails to succeed after 9=3×3 retries. With the /I:num option, QUIKCOPY will ignore an error if it fails after 3×num retries. Here, num is in the range 1 to 100.

NB Note that only one of these parameters may be specified at one time.

4.8 Cancelling Copy Process

At any stage during the copying process, QUIKCOPY may be terminated by pressing , or one of the Break keys, ,  or . Often, one may be prompted whether or not to continue copying other diskettes.

It may be necessary to press either of these key-combinations if QUIKCOPY is taking too long to copy a particular (often damaged) diskette. The Abort, Ignore, Retry, Fail message is not displayed while QUIKCOPY is running. After pressing one of these keys, a message indicating that a certain diskette was not copied will be displayed, and will ask whether or not to continue copying with other diskettes. Pressing  will continue; otherwise, pressing , , , or one of the Break keys, ,  or , will stop the copying process.

Note that this message may also be displayed if an error, such as the destination diskette being write-protected, occurs. A similar response will be required.

5. TRULY WILD WILD-CARDS

As of v3.40, QUIKCOPY supports an extensively improved set of wild-cards. The new wild-cards are backward compatible with those supported by DOS, and are a superset of that included in 4DOS and NDOS. Many of the improvements of this wild-card system are straight-forward, however there are equally as many features of this system are quite complicated. This section will try to describe this system, and try to eliminate any misunderstandings as to how to use them.

The best way to describe and explain the use of this wild-card system, is by way of example. Users already familiar with the wild-cards of 4DOS and NDOS will find the transition process a breeze. Others users will quickly learn how to use them with experience, and the help of the examples included here. This section essentially teaches by example.

Note that any combination of what is discussed in the following four sub-sections may be used.

5.1 *Truly Wild Wild-Cards Explained*

EXAMPLE 1:

Suppose you wish to copy all files containing S somewhere in the name-part of the filename, simply use:

S.*

as a source parameter. Files such as S.TXT, RJS.EXE and MYSALES.WKS will all be accepted.

EXAMPLE 2:

Suppose you wish to copy all files that have the phrase COPY somewhere in the name-part of the filename. In this case you would use:

COPY.*

as a source parameter.

EXAMPLE 3:

Suppose you wish to copy all files starting with either A, B, C or D. In this case you could use:

[A-D]*.*

as a source parameter. This is equivalent to using:

A*.*;B*.*;C*.*;D*.*

as the source parameter. (These multiple source parameters are discussed in **Section 5.3.**)

EXAMPLE 4:

If you wish to copy all files starting with A, E, I, O or U, then use:

```
[AEIOU]*.*
```

as a source parameter. Again, this is equivalent to saying:

```
A*.*;E*.*;I*.*;O*.*;U*.*
```

EXAMPLE 5:

Combinations of the above are also allowed. Suppose you wish to copy all files that have R, J or S somewhere in the extension-part of the filename. Simply use:

```
*.*[RJS]*
```

as a source parameter. This is equivalent to:

```
*.*R*;*.*J*;*.*S*
```

EXAMPLE 6:

As well as being able to specify certain characters that must appear somewhere in a filename, you may also specify certain characters that do not appear somewhere in a filename, by using the “!” character (exclamation mark) in front of them.

Suppose you wish to copy all files that do not start with A or B, then you would use:

```
[!A-B]*.*
```

or

```
[!AB]*.*
```

as a source parameter. Notice that these two parameters are equivalent. Probably, the latter (shorter) parameter would be preferred.

EXAMPLE 7:

Suppose you wished to copy all files whose name-parts were three characters long, and start with AB. One may think of using

```
AB?.*
```

yet this will copy files of the type AB.* as well. Instead, one should use:

```
AB[?].*
```

as a source parameter.

5.2 Excluding Files

As of v3.40, QUIKCOPY allows one to exclude certain files, while copying. The sequence [!“...”] should be used. Again, examples follow:

EXAMPLE 1:

Suppose you wish to copy all files, except * .BAK files. You would use:

```
[!"* .BAK"]
```

as a source parameter.

EXAMPLE 2:

Suppose you wish to copy all files that do not start with R, J or S, then you would use:

```
[!" [RJS]*.*"]
```

as a source parameter.

EXAMPLE 3:

One should be careful how you specify files that should be excluded. Often what seems intuitively correct at first will not work.

Suppose you wish to copy all files that do not contain an S, anywhere, in the name-part of the filename. One might think that

```
* [!S]*.*
```

will “do the trick”, however one will copy files containing one or more Ss in the name-part of the filename. In fact, the only files that will be excluded are files of the form SSSSSSSS.*. Instead, one should use:

```
[!"*S*.*"]
```

as a source parameter. What at first may have seemed to be intuitively correct, is in fact not. The former means that any file which does not contain an S in one of the name-part positions in the filename may be copied, whilst the latter means that only files that don't contain a single S in the name-part will be copied.

EXAMPLE 4:

It is important to note that these wild-cards and file-exclusions apply only to the filename part of a path. Thus, one should not include wild-cards and/or file-exclusions within the directory part of a path. This has particular implications with file-exclusions.

Suppose you are copying all your source-code for programs you have written over the years onto diskette. However, you have long since realized that BASIC is not such a useful programming language, and would rather not copy those source-files. You have your files interspersed within many subdirectories in the C:\MYSOURCE subdirectory. You only ever used BASIC when writing the MYAPP application, which has its source-code in the C:\MYSOURCE\MYAPP subdirectory. This subdirectory does, however, contain some assembler code that you may find useful with future programs you write in Pascal or C++. So, it seems a good choice would be to specify

```
C:\MYSOURCE\*.*; [!"C:\MYSOURCE\MYAPP\*.*BAS"]
```

as the source parameter. However, there are no files that have backslashes (“\”) in their names, so the *.BAS files will end up being copied. Instead, you should use:

```
C:\MYSOURCE\*.*;C:\MYSOURCE\MYAPP\["*.BAS"]
```

as the source parameter.

5.3 Multiple Source Files

With the release of v3.40 of QUIKCOPY, you may now specify multiple source parameters. By simply separating each source “mask” with a semi-colon (“;”), you can have multiple source files.

EXAMPLE 1:

Suppose you wish to copy all *.EXE, *.BAT and *.COM files. The follow source parameter would do the trick:

```
*.EXE;*.BAT;*.COM
```

NB Note that no spaces are allowed between masks making up the source parameter.

EXAMPLE 2:

One may take advantage of being able to specify multiple source files, and apply it copying files from multiple disk-drives.

Suppose you wish to copy of *.DOC files from drives C: and D: to current subdirectory on drive E:. The command-line for QUIKCOPY would look something like the following:

```
QUIKCOPY C:*.DOC;D:*.DOC E: /S/L
```

5.4 Extended Parent Subdirectory Names

As well as having more powerful wild-cards, QUIKCOPY also has extended parent-subdirectory specifications. The extended parent-subdirectory names are the same as those provided by 4DOS and NDOS.

Essentially, what is involved is adding an extra period (“.”) for each “ancestral” subdirectory. The following table outlines how it works:

SUBDIRECTORY	Extended	DOS
Current	.	.
Parent
Grand-parent\..
Great-grand-parent\..\..
Great-great-grand-parent\..\..\..

The above table only includes a few parent-subdirectory levels. Naturally, one may extend them even further. The left-hand column, labelled “Extended”, contains the

extended parent-subdirectory names, while the right-hand column, labelled “DOS”, contains the equivalent conventional parent-subdirectory names supported by DOS.

NB QUIKCOPY still supports the conventional parent-subdirectory naming, provided by vanilla DOS.

If one specifies too many parent-subdirectories, **QUIKCOPY** will use the root directory of a particular drive.

6. QUIKCOPY PARAMETER LISTING

This section contains a listing of the various parameters available in QUIKCOPY, listed in alphabetical order. Short descriptions of each parameter are given. For more detailed discussions on these parameters, refer to **sections 3 and 4**.

/A (file copying)

Specify that QUIKCOPY copy (or move) all files and subdirectories, including those with their hidden and system attributes set.

/B (diskette copying)

Specify that QUIKCOPY may copy a single diskette to a batch of destination diskettes.

/B (file copying)

Specify that QUIKCOPY may copy (or move) files from a batch of multiple diskettes.

/BIOS (diskette copying)

Specify that QUIKCOPY perform diskette copying by going through BIOS, and not by directly accessing the diskette-drives. See **Section 4.1** for a discussion on when this parameter should be used.

/C (file copying – diskette filling)

Indicates that during maximal diskette filling, files too large to fit on a diskette should be split across multiple diskettes.

/D (diskette copying)

Invoke the diskette copying mode of QUIKCOPY.

/DELAY and /DELAY:ms (file copying)

Slow down file copying (or moving). (See **Section 3.2.6**.)

/E (file copying)

Copy (or move) all subdirectories, including empty subdirectories.

/F (diskette copying)

Specify that QUIKCOPY always format destination diskettes before writing to them.

/F (file copying)

Indicate that QUIKCOPY should maximally fill a number of diskettes with the contents of one or more subdirectories.

/HELP and /?

When either of these parameter are specified, QUIKCOPY will display a short summary of all the parameters and briefly describe their functions. Note that only one of these parameters should be used at one time, and that no other parameters are allowed when asking for help.

/I and /I:num (diskette copying)

Specify that when copying diskettes, QUIKCOPY should ignore errors after a number of failures. (See **Section 4.7.**)

/IP (file copying – diskette filling – splitting files)

Tell QUIKCOPY to copy the QC_PASTE.EXE utility onto diskettes when splitting large files across multiple diskettes.

/L (file copying – subdirectories)

Specify that QUIKCOPY is to lock the destination subdirectory, and copy the files to that subdirectory.

/M (file copying)

Indicate that instead of copying files, QUIKCOPY should rather move them from source to destination.

/M+ and /M- (diskette copying)

Specify whether or not QUIKCOPY is to copy multiple diskettes. (See **Section 4.2.**)

/N (diskette copying)

Indicate that QUIKCOPY may not utilize disk-space as virtual memory when copying diskettes.

/NJ (file copying)

Indicate that QUIKCOPY should ignore the special status of QC_PASTE.EXE, stub-files and cut-files (see **Section 3.2.7**), and simply copy (or move) them.

/O, /OY and /-Y (file copying)

Ensure that QUIKCOPY perform overwrite checking of files on the destination. Default answer to prompts will be “yes”.

/ON (file copying)

Ensure that QUIKCOPY perform overwrite checking of files on the destination. Default answer to prompts will be “no”.

/ONA (file copying)

Specify that QUIKCOPY never overwrite files that already exist on the destination.

/OYA and /Y (file copying)

Indicate that QUIKCOPY should always overwrite files that already exist on the destination.

/P, /PY and /PN (file copying)

Prompt user for permission to copy (or move) each file that matches the source specification. (See **Section 3.2.4.**)

/S (file copying)

Copy (or move) subdirectories, but exclude subdirectories that are empty.

/VF, /VW, /VA and /VN (diskette copying)

Specify level of diskette verification required. (See **Section 4.5.**)

/W (file copying – diskette filling)

Indicate that QUIKCOPY should wipe the contents of each and every diskette used when maximally filling diskettes, before writing to them.

7. TIPS AND TROUBLE SHOOTING

This section will discuss a few tips that are advantageous to adopt when using QUIKCOPY, in that they will improve copying performance or make the process of copying more fluent. Also, various problems that may arise are also discussed, and possible solutions are given.

This section will be presented as a number of questions and answers. The question part will ask how to overcome a problem or how to perform a particular task, and the answer part will give a possible solution. Many of these questions are commonly posed by current users of QUIKCOPY.

Q How do I copy to/from a diskette image file?

A Simply specify the filename as the source or destination option.

Q Can QUIKCOPY copy non-standard diskette formats? If so, since when?

A Yes. Since v3.00.

Q Do I need FDREAD, SMAX or similar programs when copying non-standard diskettes?

A No. QUIKCOPY will copy these diskettes without the need for special TSRs or device-drivers.

Q How do I invoke QUIKCOPY's diskette copying facilities?

A Simply add the /D parameter to the command-line.

Q Do I need to enter QUIKCOPY A: A: /D when copying diskettes in drive A only, as with DOS's DISKCOPY?

A No, if you only specify a source parameter, then QUIKCOPY will assume that the destination is the same drive as the source drive. This, of course, does not hold for diskette image files.

Q Is there any specific order in which I should specify command-line parameters?

A No, any order may be used. Just ensure that a source parameter is given. Also, if a destination parameter is specified, ensure that it is listed after the source parameter.

Q Can QUIKCOPY copy 1.44M diskettes onto 1.2M diskettes? If so, since when?

A Yes. Since v3.00.

Q I ran QUIKCOPY the other day, and it said that it was possibly infected by a virus, and that I should contact you. What is wrong? What should I do?

A Firstly, something or someone has tried to change QUIKCOPY. Do not execute this copy of QUIKCOPY. Ascertain whether or not QUIKCOPY is indeed infected. Clean-up your system, removing the virus. Then, re-install an uninfected (or unchanged) version of QUIKCOPY.

Note that many anti-virus software programs do not restore a previously infected version of QUIKCOPY to its original state. That is why you need to re-install QUIKCOPY.

Q I ran QUIKCOPY, and it terminated with the following message:

ERROR - Could not install file manager!

What is the problem?

A **Section 2.1.1** has an extensive discussion on this problem. Refer to this section on how to circumvent this problem.

Q I ran QUIKCOPY, and it terminated with the following message:

ERROR - Problem with system timer.

What is the problem?

A When it first starts executing, QUIKCOPY calibrates itself to the speed of your system. This calibration depends upon the system timer. If a faulty or incompatible system timer is present in the system, then QUIKCOPY will not work with your computer. This error should never occur. However, if it does occur, retry running QUIKCOPY, to see if it was only a temporary error.

Q The documentation mentions that the performance of the maximal diskette filling process may be improved. How do I go about doing this?

A It is highly recommended that you run a disk-caching program on your system. Not only will QUIKCOPY's performance be improved, but that of your entire system in general. Choice of a disk-cache is rather personal. However, a good choice would be a disk-cache that has *write-back* or *write-delay* facilities. Most good modern disk-caches have this ability. Enabling this feature of your disk-cache will greatly improve disk-performance. This is where the benefits for QUIKCOPY arise.

When filling diskettes (especially), the presence of a disk-cache that does write-caching greatly improves QUIKCOPY's performance, especially if filling diskettes with the contents of multiple subdirectories. QUIKCOPY performs disk-cache flushing, so no data-loss will arise (unless of course external phenomena, such as power-outs, occur).

File-moving will also be somewhat faster if a disk-cache capable of caching disk-writes is present in the system, especially if moving files on the same drive.

Q Do you have any special recommendations that would further improve the diskette filling process?

A Yes, one. It may prove useful, if you plan to fill diskettes of varying sizes, that you order these diskettes in decreasing order of capacity. Thus, a smaller number of diskettes will be required, and also, less files will (possibly) need to be split across multiple diskettes.

Q In the documentation, it is mentioned that QUIKCOPY could be used as a backup and restore system. How would I go about doing this?

A Simply create two batch files, say BACKUP.BAT and RESTORE.BAT, that can perform this process. The former would utilize QUIKCOPY's diskette filling capabilities (including large-file splitting) along with subdirectory inclusion to backup the data to multiple diskettes. The latter could simply utilize QUIKCOPY's multiple diskette batch copying capability to restore the data.

See the files BACKUP.BAT and RESTORE.BAT included with this program. These are examples of how one could perform this process.

Q Do you have any plans for a menu-driven version of **QUIKCOPY**?

A Yes. It will be called **QUIKCOPY Professional (QCPRO)**. It should be released sometime this year (1994). It will be commercial though. At the same time when **QCPRO** becomes available, **QUIKCOPY v4.00** will be released. This will still be released as shareware. **QUIKCOPY** will be a heavily watered-down version of **QCPRO**. Many of **QCPRO**'s features will either be disabled in the shareware edition of **QUIKCOPY**, or simply not be included. **QCPRO** will be available for both the **DOS** and **Windows** platforms.

8. ACKNOWLEDGEMENTS

This documentation refers to other software and hardware. The author and RJS Software acknowledge the copyright of the respective authors and companies that hold the copyrights for this other software/hardware.

The following is a list of various software/hardware mentioned in this documentation along with the authors/companies that hold the copyrights.

Note that the author and RJS Software acknowledge the copyrights of any software/hardware not included below. Note also that the information in this list may not be 100% correct.

- The word “DOS”, used in this documentation, refers to Microsoft's MS-DOS or IBM's PC DOS.
- MS-DOS, SMARTDRIVE and Microsoft Windows are trademarks or registered trademarks of **Microsoft Corporation**.
- DR DOS is **Digital Research's** operating system, now owned by **Novell**.
- The terms PC, XT, AT, PC DOS and IBM are trademarks or registered trademarks of **International Business Machines**.
- 4DOS is a command processor replacement for DOS. It is copyright **JP Software Incorporated**.
- NDOS refers to Norton DOS, supplied with Norton Utilities (under license from **JP Software**), and distributed by **Symantec**.
- PC-CACHE is a disk-cache included with PC Utilities, copyright **Central Point Software**.
- HYPERDISK is a disk-cache, copyright by **HyperWare**.
- DESQview is copyright **Quarterdeck Office Systems**.
- Intel and Pentium are trademarks or registered trademarks of **Intel Corporation**.
- MAXI Form and SMAX are copyright **Herne Data Systems**.
- FDFormat and FDREAD are copyright **Christoph Hochstätter**.
- LZEXE is copyright **Fabrice Bellard**.
- PKLITE is a trademark of **PKWARE Incorporated**.

Finally, QUIKCOPY, QUIKCOPY Professional, QCPRO, RJS Software and the *RJS Software logo* are trademarks of **RJS Software**.

9. REGISTRATION AND CORRESPONDENCE

This section contains information about registering QUIKCOPY, your rights as a user of this software, how to contact the author or RJS Software, as well as other important information regarding QUIKCOPY.

9.1 QUIKCOPY is Shareware

QUIKCOPY is **not** a free program. It is being released as shareware. This means that you are allowed to try out QUIKCOPY and its various features for a limited amount of time, in which you may evaluate the suitability of this software to your use.

QUIKCOPY has not been disabled in any way in the shareware version. It just contains a message that reflects that it is not free, and must be registered. One is granted a 14 day period in which to evaluate the suitability of QUIKCOPY to one's needs. If you use QUIKCOPY for a period longer than this evaluation period, it is assumed that QUIKCOPY is of use to you, and that you will register it.

One may register QUIKCOPY by sending a fee of R25 to the address below, if you are a resident of South Africa, or one of its neighbouring states. Users outside of South Africa, may register QUIKCOPY by sending \$20 US to the address given in **Section 9.3**. (A registration form is included with this program.) All postage and packaging is included in this fee.

Registration may be paid in the form of postal order, cheque or by credit card. If paying by credit card, registration by FAX is allowed. Please print out and fill in registration form, which is contained in the file QUIKCOPY.ORD.

Registered users will receive the latest version of QUIKCOPY, without the unregistered message. After registering, one may upgrade to the next five versions of QUIKCOPY for free, and there-after one is entitled to an upgrade price of R10 or \$10 US for the next three versions. Registered users will also receive a unique serial-number that they must quote on all correspondence after they have received their registered version of QUIKCOPY.

9.2 DISCLAIMER

QUIKCOPY is provided as is. It has been tested extensively, and has proven to fulfill the task it has been designed to do. The author or RJS Software do not accept any responsibility for damage that may be incurred out of the use of, abuse of, or the inability to use the features provided by this program.

This program may not be modified, reverse-engineered, hacked (!!!), etc. in any way whatsoever. It may not be sold for profit by any party, without prior written permission from the author, Roland Skinner, and RJS Software.

The unregistered version may not be distributed without this documentation (QUIKCOPY.DOC), the order-form (QUIKCOPY.ORD), the history-list (HISTORY.LST), the thank-you list (THANKYOU.LST), or any other file originally included with this software.

This program does perform an integrity check on execution. "Modified" versions could have disastrous effects on a system – BE WARNED! Do not attempt to compress this file, in executable form, using programs such as LZEXE or PKLITE – QUIKCOPY will refuse to work.

People using QUIKCOPY have a 14 day evaluation period in which to use the unregistered version. People using QUIKCOPY in a commercial environment must register QUIKCOPY in order to use it. (Cheaper rates are available for registering multiple copies of QUIKCOPY; contact the author or RJS Software for more information.)

By having read this, one agrees to abide by the above. By using this software one acknowledges having read this section, and agree to the clauses above.

9.3 Contacting RJS Software or the Author

Any correspondence may be forwarded to the address below.

Send your recommendations, bug-reports, requests, registrations, or any other correspondence to Roland Skinner, or RJS Software, at:

P.O. Box 14134
Bredell
1623
Republic of South Africa

The above address is not a residential address; only a box number.

FAX support is provided by calling the following number:

+27-11-792-4874 (International)
(011) 792-4874 (within South Africa)

Anybody who is the first to recommend a feature that is later added to QUIKCOPY, or is the first to report an error in the working of QUIKCOPY, will have his/her name added to the THANK YOU list accompanying this documentation.

Users requesting faster replies, etc. may send an Email message to Roland Skinner in one of the echomail conferences in South Africa. I recommend either RsaSOFT or TP_Shareware. In the world-wide Ilink network, I may be contacted in the Ilink SHAREWARE conference.

I may also be contacted privately through FIDONET netmail or through Internet Email. Preferably, try to reach me via the Internet.

I may be contacted at the following Email addresses (listed in order of preference). Also, the echomail conferences are included here. The FAX number is also repeated here.

Email:

Internet: roland.skinner@leclub.proxima.alt.za
roland@concave.cs.wits.ac.za
roland.skinner@p1.f12.n7107.z5.fidonet.org
FIDONET: Roland Skinner @ 5:7107/12.1

Echomail:

RsaNET: RsaSOFT conference (Roland Skinner)
TopNet: TP_Shareware conference (Roland Skinner)
llink: SHAREWARE conference (Roland Skinner)

FAX:

+27-(0) 11-792-4874

I trust that you'll find **QUIKCOPY** to be a useful utility. Hopefully, you too, as many other users already have will benefit from the various features **QUIKCOPY** provides.

Yours faithfully,

Roland Skinner