FileFunctions

COLLABORATORS			
	TITLE : FileFunctions		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		February 12, 2023	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

FileFunctions

1.1 Chapter 3 - File Functions

Previous Chapter: Next Chapter:
2. Files

CHAPTER 3 - FILE FUNCTIONS

1.2 Create Directories

CREATE DIRECTORIES

To create directories you use the function CreateDir() only need to give the function a string with the name of the new directory (including any necessary path), and the directory will automatically be created. If the directory could successfully be created an exclusive lock to the new directory is returned, else if there was an error and the directory could not be created NULL is returned.

Since the directory is automatically locked if it was created

```
you have to unlock it yourself as soon as you do not need the
lock any more.

If there already exist a directory with the same name the
function will fail.
```

1.3 Delete Files and Directories

```
DELETE FILES AND DIRECTORIES

When you want to delete a file or directory you should use the
 DeleteFile()
string with the name of the file or directory you want to delte
(including any necessary path). If the file/directory could be
deleted TRUE is returned, else FALSE is returned (file could
not be deleted).

Note that you can only delete a directory which is empty!
```

1.4 Rename Files and Directories

```
RENAME FILES AND DIRECTORIES

When you want to rename a file or directory you should use the
 Rename()
to other directories on a volume by simply renaming the path
accordingly. For example:

  /* Move a file inside a volume: OK! */
  Rename( "df0:Documents/Sale.doc", "df0:Letters/Sale.doc" );

We have now moved the file "Sale.doc" from the directory
"Documents" to directory "Letters". Note! You can not rename a
file from one volume to another, you can only move the file or
directory inside the volume. Here is an example that will not
work:

  /* Try to move a file to another volume: INCORRECT! */
  Rename( "df0:Documents/Sale.doc", "df1:Letters/Sale.doc" );

If you move a directory all objects (files and subdirectories)
will also be moved accordingly.
```

1.5 Rename Volumes (Relabel)

```
RENAME VOLUMES (RELABEL)

When you want to rename a volume you can not use the previously
discussed  Rename()
function called  Relabel()
```

```
2. Before you may use this function you must therefore make
sure that the user really has dos version 36 or higher!
```

1.6 Attach Comments to Files and Directories

```
ATTACH COMMENTS TO FILES AND DIRECTORIES

AmigaDOS allows you to set a comment on a file or directory.
The comment can give the user a brief description of what the
file contains, or who created it etc... This comment is very
handy to use when you want to give some extra information about
the file, and since you can connect a comment which is up to 80
characters long even rather long descriptions can be used.

The user can display a comment on a file or directory by using
the Shell (CLI) command "List":

  1.Prog:>
  1.Prog:> List df0:SourceCodes
  Directory "df0:SourceCodest" on Monday 15-Mar-93
  Miniblast.c              34155 ----rwed 12-Jan-93 23:35:20
  : This version can be used on all Amigas! Still a bug on line 365!!! -AB-
  Macroblast.c            299128 ----rwed 16-Jan-93 03:35:12
  : Connects fine, prob. with unknown PubScreen, otherwise OK! -AB-
  LockSystem.c             24899 ----rwed 04-Feb-93 01:45:34
  : Has to be updated before use! Line 512 & 525, ok code, -AB-
  and so on...

To set a comment on a file or directory you should use the
 SetComment()
```

1.7 Alter the Protection Flags

```
ALTER THE PROTECTION FLAGS

All files and directories have a field of bits which is called
"the protection bits". The bits can be turned on or off and it
will affect other programs that may use the files. The most
commonly used bit is bit 0 which tells AmigaDOS if the file or
directory may be deleted or not. This is the "delete" bit, and
since it is so commonly used the whole collection of bits has
been called "the protection bits" although the other bits have
nothing to do with protecting the file from being deleted.

The user can alter the bits with help of the Shell (CLI)
command "Protect". Here is the list of protection flags the
user can use (each flag will alter its corresponding bit):

  Flag Description
  ------------------------------------------------------------
    s  The file is a "script" and contains Shell commands. If
       the user enter the name of a file with this flag set
```

will the file automatically be executed as a script. The user do not have to add the Shell command "Execute" in front of the file name.

p The file is "pure" and can therefore be made "resident". If this falg is not set the Shell command "Resident" will fail.

a The file has been archived. Everytime a file has been modified this flag is removed. The flag is usually used by backup programs which set this flag each time the file has been archived.

r The file can be read by other programs. If this flag is not set the file can not be read.

w The file can be altered by other programs. If this flag is not set the file can not be altered.

e The file is "executable" (a normal program). If this flag is not set Shell (CLI) will refuse to start the file as a program.

d The file/directory can be deleted. If this flag is not set the file can not be deleted.

Most of these flags are only useful for files, but it is possible to use all of then on directories too even if most of them will not have any effect on the directory.

Here is an exapmple on how the user can protect a file from being accidentally deleted:

```
1.Prog:>
1.Prog:> Protect df0:Important.dat SUB d
```

To see which flags are set or not the user can call the Shell (CLI) command "List":

```
1.Prog:>
1.Prog:> List df0:
Directory "df0:" on Monday 15-Mar-93
Important.dat           128722 ----rwe- 21-Feb-93 02:12:18
NotSoImportant.dat      103677 ----rwed 21-Feb-93 02:12:25
and so on...
```

As you can see has the "d" flag been removed, and the file can then not be deletet. The only way to delete the file is to add the "d" flag again and then delete the file.

If the flag is used by AmigaDOS depends on which version the user has:

---

s Used by the Shell & CLI (WB 1.3 or higher). If the flag is set the file will be executed as a script. (You do no need to use the Shell command "Execute".)

```
    p  Used by Shell's (CLI's) "Resident" command (WB 1.3 or
       higher). If the flag is set the file (program) can be
       made "resident".
    a  Used by all AmigaDOS versions and the FFS. Every time
       the file has been altered this flag is removed.
    r  Used by AmigaDOS V36 or higher and by the FFS. If it is
       not set the file can not be read.
    w  Used by AmigaDOS V36 or higher and by the FFS. If it is
       not set the file can not be altered.
    e  Used by the Shell & CLI (all AmigaDOS versions and the
       FFS), if the flag is not set Shell and CLI refuses to
       start it.
    d  Used by all AmigaDOS versions and the FFS. If it is not
       set the file can not be deleted.
  ------------------------------------------------------------

To alter the protection bits from C you should use the function
 SetProtection()
```

1.8 Examples

```
EXAMPLES

Example 1:  Read!    Run!    Edit!
  This example demonstrates how to create a directory called
  "MyDirectory" on the RAM disk.

Example 2:  Read!    Run!    Edit!
  This example demonstrates how to rename files and
  directories. It will rename a file on the Ram disk called
  "HighScore.dat" to "Numbers.dat". It will also rename the
  directory the previous Example created ("MyDirectory") to
  "NewDirectory". (Please  run Example 1  in the  Files
  chapter to create the "HighScore.dat" file before you run
  this example.)

Example 3:  Read!    Run!    Edit!
  This example demonstrates how to delete files and
  directories. It will delete the file and directory which we
  renamed in the previous example.

Example 4:  Read!    Run!    Edit!
  This example demonstrates how to attach a comment to a file.
  The commend will be attached to a file on the Ram disk called
  "HighScore.dat". (Please  run Example 1  in the  Files  chapter
  to create this file before you run this example.) Too see the
  comment you can use the Shell (CLI) command "List". Simply
  open a Shell window and type  List RAM: .

Example 5:  Read!    Run!    Edit!
  This example demonstrates how to alter the protection flags
  on a file. The file we used in the previous example
  "HighScore.dat" will be protected and we will then try to
  delete it (unsuccessfully). We will then unprotect the file
```

```
and try to delete it again (this time successfully).
```