MRB_Filters

COLLABORATORS			
	TITLE : MRB_Filters		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		December 11, 2024	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 MRB_Filters **1**

1.1 Filter Files 1

1.2 filter file format 1

1.3 backup filter 3

1.4 compression filter 3

1.5 decompression filter 4

Chapter 1

MRB_Filters

1.1 Filter Files

The term "filter" may sound strange to you, but you've probably heard and even used it many times without giving it a second thought. Surely you've heard of the coffee filter, which keeps the coffee grounds out of your freshly brewed pot of java. Air filters keep dust and dirt out of your electronic equipment and your environment. Oil filters keep your auto's engine clean.

MRBackup employs filename filters to accomplish something quite analogous. A set of filenames is "fed into" a filter and a subset of those filenames is allowed to "pass through" it. Each filter is simply a text file containing zero or more filename patterns. Each filter (backup, compression, decompression) has a specific purpose and operates on a particular set of filenames.

Filter File Format

Backup Filter

Compression Filter

Decompression Filter

Go to Top Level

1.2 filter file format

Each filter file is simply a text file which can be created with any plain text editor or a word processor which can save plain ASCII text. They can also be created by some other program or ARexx script (hint!). The filter file may contain any number of patterns, one per line. You may place comments in the file by placing a semi-colon (;) in the first character position. Empty lines are also ignored.

Filter Patterns
Filter patterns are expressed "relative" to a device or volume. That is,

they must not include a volume or device name (the part preceding and including the colon in a full AmigaDOS filename). What they are relative to is implicit in their use. For instance, the backup filter patterns are relative to the "home" device, as are the compression filter patterns. Since decompression is performed during a restore, decompression filter patterns are implicitly relative to the "backup" device.

Filter patterns may be simple filenames, such as

```
Trashcan
Trashcan.info
```

or they may be quite exotic and complex patterns with "wildcard" notation, character class specifications, etc. Here is the definition of special characters which can be used in a filter pattern:

Character	Meaning
?	Match any single character.
%	Match the null string.
#<p>	Match zero or more occurrences of pattern <p>.
*	Match any pattern (same as #?).
<p1><p2>	Match pattern <p1> followed by pattern <p2>.
( )	Parentheses group patterns together.
(<p1>|<p2>)	Match if either <p1> or <p2> match (parentheses are required.).
[ ]	Character class (ex: [a-z] or [0-9]).
~<p>	Negation: match anything BUT pattern <p> Example: "~*.info" means all files except those ending in ".info" (quotes for illustration only)
'	Escape next special character. Useful for filenames which contain any special characters above.

The most common mistake that you are likely to make when creating your filter patterns is to omit the "leading context" from your patterns. For instance, if you want to omit all files named "junk.txt" from an operation, you must remember that the simple filename is actually part of a larger specification which includes all higher level directory names. Thus, to omit all files named "junk.txt", we might use the following pattern:

```
(junk.txt|#?/junk.txt)
```

This is a pattern grouping which recognizes "junk.txt" at the top level of a volume or (you can equate the vertical bar | to the word "or") at any level in the directory hierarchy. The pattern

```
#?junk.txt
```

is not "safe" since it will match ANY sequence of characters preceding "junk.txt" (somejunk.txt, myjunk.txt, morejunk.txt, etc.) which is not be what we wanted.

1.3 backup filter

The Backup Filter is used to assist in the selection of files that are to be copied during a backup operation. When MRBackup performs its initial scan, the backup filter is applied to each file or directory name as it is encountered.

The Backup Filter has a dual personality. By default, its patterns are used to exclude selected files from a backup. However, there are two special patterns which change the meaning of the Backup Filter patterns. These patterns are

```
:INCLUDE:   and
:EXCLUDE:
```

If the :INCLUDE: pattern appears in the Backup Filter file, subsequent patterns will be used to include files in the backup. Only files which match these patterns will be included in the backup. This can be a useful mechanism for backing up "disjoint" directory hierarchies without having to provide many exclude patterns.

If the :EXCLUDE: pattern appears in the Backup Filter file, subsequent patterns will be used to exclude files from the backup. If the :INCLUDE: pattern is also present in the filter file, the exclude patterns will be applied only to the files which satisfied the include patterns. That is, the include patterns take precedence, regardless of the appearance order of :INCLUDE: or :EXCLUDE:.

1.4 compression filter

The Compression Filter is employed during a backup operation to inhibit file compression on certain files. It has no effect if the Compression gadget has been set to None.

Over time, you will notice that certain files have a tendency to expand, rather than compress, when subjected to MRBackup's compression algorithm. Such strange behavior! The compression algorithm takes advantage of the fact that most files contain data whose values are not evenly distributed. There tend to be many redundant data patterns which can be represented by fewer bits. However, certain files, such as programs, animations, graphics images, etc., do not compress well since they already contain very random data patterns. MRBackup will detect this condition, abandon compression for that file and perform a straight copy. Unfortunately, this results in a waste of time, since a significant portion of the file may have been compressed before the condition was detected.

MRBackup helps you alleviate this situation by providing you with a compression filter. When you detect files for which compression is a

problem, enter the appropriate patterns into your compression filter file and MRBackup will cease trying to compress them. If you use a reasonable naming convention for certain classes of files (e.g. <file>.ilbm for IFF bitmap files), you can easily omit whole classes of files from compression.

1.5 decompression filter

While file compression is a nice backup feature, you may also want to maintain certain files on your hard drive in a compressed state. If you perform a restore operation with decompression enabled, however, files that you wish to remain compressed will be decompressed. The decompression filter allows you to specify the files which you would like to restore in their compressed state. That is, patterns in the decompression filter inhibit file decompression.