



Red Hat Enterprise Linux Release 9.2 Manual Pages on 'fnmatch.3am' command

\$ man fnmatch.3am

FNMATCH(3am) GNU Awk Extension Modules FNMATCH(3am)

NAME

fnmatch - compare a string against a filename wildcard

SYNOPSIS

```
@load "fnmatch"
```

```
result = fnmatch(pattern, string, flags)
```

DESCRIPTION

The `fnmatch` extension provides an AWK interface to the `fnmatch(3)` routine.

It adds a single function named `fnmatch()`, one predefined variable (`FNM_NOMATCH`), and an array of flag values named `FNM`.

The first argument is the filename wildcard to match, the second is the filename string, and the third is either zero, or the bitwise OR of one or more of the flags in the `FNM` array.

The first argument is the filename wildcard to match, the second is the filename string, and the third is either zero, or the bitwise OR of one or more of the flags in the `FNM` array.

The return value is zero on success, `FNM_NOMATCH` if the string did not match the pattern, or a different non-zero value if an error occurred.

The return value is zero on success, `FNM_NOMATCH` if the string did not match the pattern, or a different non-zero value if an error occurred.

The return value is zero on success, `FNM_NOMATCH` if the string did not match the pattern, or a different non-zero value if an error occurred.

The return value is zero on success, `FNM_NOMATCH` if the string did not match the pattern, or a different non-zero value if an error occurred.

The flags are follows:

`FNM["CASEFOLD"]`

Corresponds to the `FNM_CASEFOLD` flag as defined in `fnmatch(3)`.

`FNM["FILE_NAME"]`

Corresponds to the `FNM_FILE_NAME` flag as defined in `fnmatch(3)`.

`FNM["LEADING_DIR"]`

Corresponds to the `FNM_LEADING_DIR` flag as defined in `fnmatch(3)`.

`match(3)`.

FNM["NOESCAPE"]

Corresponds to the FNM_NOESCAPE flag as defined in `fnmatch(3)`.

FNM["PATHNAME"]

Corresponds to the FNM_PATHNAME flag as defined in `fnmatch(3)`.

FNM["PERIOD"]

Corresponds to the FNM_PERIOD flag as defined in `fnmatch(3)`.

NOTES

Nothing prevents AWK code from changing the predefined variable FNM_NO?MATCH, but doing so may cause strange results.

EXAMPLE

```
@load "fnmatch"
...
flags = or(FNM["PERIOD"], FNM["NOESCAPE"])
if (fnmatch("*.a", "foo.c", flags) == FNM_NOMATCH)
    print "no match"
```

SEE ALSO

GAWK: Effective AWK Programming, `filefuncs(3am)`, `fork(3am)`, `in?place(3am)`, `ordchr(3am)`, `readdir(3am)`, `readfile(3am)`, `revoutput(3am)`, `rwarray(3am)`, `time(3am)`.
`fnmatch(3)`.

AUTHOR

Arnold Robbins, arnold@skeeve.com.

COPYING PERMISSIONS

Copyright ? 2012, 2013, 2018, Free Software Foundation, Inc.

Permission is granted to make and distribute verbatim copies of this manual page provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual page under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual page into another language, under the above conditions for modified

versions, except that this permission notice may be stated in a trans?

lation approved by the Foundation.

Free Software Foundation

Feb 21 2018

FNMATCH(3am)