

Using pattern matching (wildcards) in SafeZone settings

This Symantec AntiVirus 10.2 for Mac update includes functionality in Auto-Protect to allow matching path names against “patterns”, i.e. strings containing special characters that match multiple file names. In particular, the special characters recognized are:

- * Matches zero or more characters (all characters match)
- ? Matches a single character (all characters match)
- [] Matches a single character against a list of characters and/or ranges of characters
- ^ Matches a character other than character or range following (used with [])

A string is considered to be a possible pattern if it contains at least one of the 3 characters *, ? or [.

Examples:

Pattern: /Users/*/Preferences/*

Path names:

- /Users/Lee/Preferences/com.symantec.plist → matches
- /Users/Lee/Preferences/P1/com.whatever → matches
- /Users/Bill/Preferences → doesn't match (last “/” required)
- /Users/Bill/Preferences/ → matches

Pattern: /[A-Z]?

Path names:

- /A1 → matches
- /h+ → matches (case-insensitive, see Note 1)
- /21 → doesn't match (“2” is not in range A-Z)
- /ABB → doesn't match (too many characters)

Pattern: [a-hm-wz]

Path names:

- g → matches
- j → doesn't match (“j” is between “h” and “m”)
- q → matches
- Z → matches (case-insensitive)

Pattern: [^A-Z^1-5]*

Path names:

- 7eee → matches
- 8 → matches
- 3abc → doesn't match (“3” is in range “1-5”)
- A → doesn't match (“A” is in range “A-Z”)
- _A → matches

Testing Patterns

A command-line tool, `filematch`, is provided to help test the use of patterns. The `filematch` tool is located in the `/Library/Application Support/Symantec/SMac/Tools/` folder. It takes two arguments: the pattern to be matched against, and a path string to be tested. Note that the pattern needs to be quoted so that the special characters are not interpreted as such by the shell (quotes should not be used when specifying patterns in the SAV/SACM interface, however). In addition to the output illustrated below, the tool returns 0 for matches and 1 for mismatches.

Examples of `filematch` usage (assumes current directory contains the `filematch` tool):

Command:

```
./filematch "/Users/*/Preferences/*" /Users/Lee/Preferences/com.symantec.plist
```

Output:

```
/Users/Lee/Preferences/com.symantec.plist matched /Users/*/Preferences/*
```

Command:

```
./filematch "/Users/*/Preferences/*" /Users/Lee/Preferences
```

Output:

```
/Users/Lee/Preferences did not match /Users/*/Preferences/*
```

Notes

1. Pattern matching is case-insensitive. For example, the pattern:

```
/abc/*
```

would be matched by:

```
/abc/d  
/Abc/d  
/abC/e  
/ABC/F
```

and so forth.

2. The ordinary containment rules for SafeZone directories do not apply when pattern matching occurs. For example,

```
/Users/Lee/Preferences/com.symantec.plist
```

does not match the pattern

`/Users/*/Preferences/`

because, although the file is contained in a user's Preferences folder, the file name does not match the pattern (adding a final "*" to the pattern would correct this).

3. In order to resolve ambiguity with file names containing special pattern-matching characters, the potential patterns are used in ordinary SafeZone matching before being checked as patterns. For example, if a user had a file named:

`/Users/[A-Z]/myfile`

where [A-Z] was literally the name of a folder in /Users, this would match the SafeZone specified by:

`/Users/[A-Z]/myfile`

through normal matching; in addition:

`/Users/A/myfile`

`/Users/B/myfile`

and so on, would match through pattern matching.

Notes about SafeZone syntax

The following behaviors should be noted when using SafeZone in SAV.

- Entering a SafeZone path that includes a filename works correctly if the file exists before the preference is loaded. If SafeZone doesn't find the file specified in the path, it treats the filename as a folder. We recommend that you do not specify a specific filename in a SafeZone path.
- Wildcard syntax for SafeZones is best used to match patterns of filenames. Using wildcards and forward slashes to specify directory names can be tricky. If you want to scan a directory and its contents, specify the name of the directory explicitly, or follow the wildcard pattern with an asterisk (*) to include all files within the directories.

The following example pattern will not recognize the last forward slash as an enclosing directory entity, and will not scan files within matching [a-b] directories:

`/Users/qa/work/tmp/testDir/foo/bar/[a-b]/`

This example will scan files within matching [a-b] directories:

`/Users/qa/work/tmp/testDir/foo/bar/[a-b]/*`