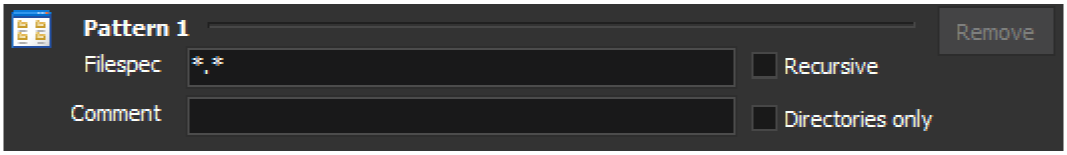
FileSet Patterns

FileSet Patterns are used to specify which files are included and excluded from the FileSet.



Pattern 1

Filespec: *.*

□ Recursive

Comment:

□ Directories only

The Filespec determines which files are added to the FileSet but files can be excluded by using exclusion patterns and filters.

The filespec can specify the following:

- Simple filespec, eg. *.cs
- Filespec including path from root directory, eg. \My project*.*
- Filespec including path occurring anywhere, eg. My project*.*

> ⚠ If you are specifying all files in a certain directory, then the *.* filespec is optional, but the directory name must end with a slash. Eg. \DIR1\ is equivalent to \DIR1*.*

Simple fileSpec

This uses standard Windows type wildcards to specify files. The asterisk "*" specifies any characters and the question mark "?" specifies any single character.

Examples

- All files: *.*
- All C# source files: *.cs
- Any file that has three characters and an extension of txt: ???.txt

FileSpec including path from root

If the filespec begins with a backslash then that sub directory must be directly under the base directory of the filespec (specified on the Define Fileset property page). If you have recursion enabled and a directory by the same name is found elsewhere in the file system but not as a direct subdirectory of the root, then the directory will not be searched. This can be thought of as simply changing the root directory for this pattern.

Example

Using two patterns, only choose files from the directories <root>\DCU\ and <root>\BAK\ even though the root directory may have many more subdirectories:

- Filespec 1: \DCU*.*
- Filespec 2: \BAK*.*

FileSpec including path occurring anywhere

If the filespec contains a directory, but does not begin with a slash and recursive is specified then the directory will be selected anywhere it is found as a subdirectory from the root. For example, consider the following file structure:

```
<root dir>

 +--- Projects

      +--- Project 1

           +--- Output

      +--- Project 2

           +--- Output
```

If you want to select all *.exe files from either of the Output directories, specify that the pattern uses recursion and use the following filespec:

- Output*.exe

Exclusion patterns

Exclusion patterns function in a very similar way to include patterns. Exclusion patterns are used to test the directory or file found by the include pattern to see if the file or directory should be included or discarded. Exclusion patterns follow the same rules as include patterns as far as the formatting of the filespec goes. Exclude patterns can be used to specify the following:

- a filespec (eg. *.exe)
- a directory from the root, eg. \Out\
- a directory anywhere, eg. Out\

Specifying a directory may also include a filespec.

Examples

All files, except executables

Include Pattern: *.*

Exclude Pattern: *.exe

All files (recursive), except any directory with the name "Output"

Include Pattern: *.*

Exclude Pattern: Output\

All files with extension .cs (recursive), except not in the Output directory as a subdirectory of the root

Include Pattern: *.cs

Exclude Pattern: \Output\