

## removeFilesFromDir

Remove files from passed directory.

### Syntax

---

```
removeStatus = removeFilesFromDir(directory)
removeStatus = removeFilesFromDir(directory, filePattern)
```

### Description

---

**removeStatus = removeFilesFromDir(directory)** removes all files that are located in the passed directory and returns a logical 1 if the operation was successful or 0 if not. The directory argument must be char vector of 1xN and valid path to a existing directory.

**removeStatus = removeFilesFromDir(directory, filePattern)** removes all files in the located directory which matches the passed file pattern. The filePattern argument must be char vector of 1xN. It is an optional argument with a default value of '\*.\*', valid file patterns can be filenames which part replace names by \* character before the dot and existing file extensions e.g. myfile\_\*.m or \*.txt and so on.

### Examples

---

```
d = fullfile('rootPath', 'subfolder')
rs = removeFileFromDir(d)
```

```
d = fullfile('rootPath', 'subfolder')
rs = removeFileFromDir(d, '*.mat')
```

### Input Arguments

---

**directory** char vector, path directory in which to scan for files with file pattern and to delete found files.

**filePattern** char vector of file pattern with extension. Default is to delete all files. Possible patterns can be passed with filename parts with start operator as place holder.

### Output Arguments

---

**removeStatus** logical scalar which is true if all files which matches the file pattern are deleted successfully from passed directory path.

### Requirements

---

- Other m-files required: None
- Subfunctions: None
- MAT-files required: None

### See Also

---

- [fullfile](#)
- [dir](#)
- [delete](#)
- [isfile](#)
- [isempty](#)
- [ismember](#)
- [mustBeFolder](#)
- [mustBeText](#)

```
function [removeStatus] = removeFilesFromDir(directory, filePattern)
    arguments
        % validate directory
        directory (1,:) char {mustBeFolder}
        % validate filePattern
        filePattern (1,:) char {mustBeText} = '*.*'
    end
    % parse pattern for dir
    parsePattern = fullfile(directory, filePattern);
    % parse directory, returns struct
    filesToRemove = dir(parsePattern);
    % delete files, tranpose to loop through struct
    for file = filesToRemove'
        % check before delete
        filePath = fullfile(file.folder, file.name);
        if isfile(filePath)
            delete(filePath);
        end
    end
    % check if dir returns an empty struct now
    check = dir(parsePattern);
    removeStatus = isempty(check(~ismember({check.name}, {'.', '..'})));
end
```