yh006150

Generated by Doxygen 1.8.17

Chapter 1

__FrontPage

Module Code: CS1PC20

Assignment report Title: CS1PC20-A- Coursework 2

Student Number: 30006150

Date: 10/01/2022

Actual hrs spent on the assignment: 30

Chapter 2

DynamicLinkCompile

```
gcc -c src/ProjectManager.c -o bin/ProjectManager

gcc -c -fPIC src/imports/create_directory.c -o bin/shared/create_directory.o

gcc -c -fPIC src/imports/create_feature.c -o bin/shared/create_feature.o

gcc -c -fPIC src/imports/remove_directory.c -o bin/shared/remove_directory.o

gcc -c -fPIC src/imports/view_directory.c -o bin/shared/view_directory.o

gcc -c -fPIC src/imports/search_directories.c -o bin/shared/search_directories.o

gcc -c -fPIC src/imports/move_directory.c -o bin/shared/move_directory.o

gcc -c -fPIC src/imports/read_and_write.c -o bin/shared/read_and_write.o

gcc -shared bin/shared/create_directory.o bin/shared/create_feature.o bin/shared/remove_directory.o bin/shared/view↩
_directory.o bin/shared/search_directories.o bin/shared/move_directory.o bin/shared/read_and_write.o -o
bin/shared/libimports.so

gcc bin/ProjectManager -Lbin/shared -limports -o bin/LinkedProjectManager

sudo mv bin/shared/libimports.so /lib
```

Chapter 3

Project Manager tool usage examples

3.1 Scenarios

3.1.1 create_project

Pre: In a folder with a file called myproject already in it

```
$ pm create_project myproject
A folder of that name already exists.  Aborting.
```

Post: No change

3.2 </blockquote>

Pre: In a folder with a folder called myproject already in it

```
$ pm create_project myproject
A folder of that name already exists.  Aborting.
```

Post: No change

3.3 </blockquote>

Pre: In a folder which does not have a pre-existing file or folder called myproject. For example, in /home/pat/cw2

```
$ pm create_project myproject
Initialized empty Git repository in /home/pat/cw2/myproject/.git/
```

Post: A new folder called myproject exists in current directory. An ls -al myproject returns:

```
total 0
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 .
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 ..
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 .git
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 bin
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 docs
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 lib
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 src
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 tests
```

3.4 </blockquote>

Pre: In a folder which does not have a pre-existing file or folder called myproject. For example, in /home/pat/cw2

```
$ pm create_project mypro ject
Wrong number of parameters, should be pm command arg
```

Post: No change

3.5 </blockquote>

Pre: In a folder which does not have a pre-existing file or folder called myproject. For example, in /home/pat/cw2

```
$ pm create_project "mypro ject"
Bad characters in folder name
```

Post: No change

3.6 </blockquote>

3.6.1 add_feature

Pre: In a folder with a folder called myproject already in it

```
$ pm add_feature feature1
A folder of that name already exists.  Aborting.
```

Post: No change

3.7 </blockquote>

Pre: In a folder which does not have a pre-existing file or folder called feature1. For example, in /home/pat/cw2/myproject

```
$ pm add_feature feature1
```

Post: A new folder called feature1 exists in current directory. An ls -al feature1 returns:

```
total 0
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 .
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 ..
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 bin
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 docs
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 lib
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 src
drwxr-xr-x 1 pat pat 4096 Nov 14 13:00 tests
```

and git branch should report feature1 master

3.8 </blockquote>

3.8.1 add_tag

Pre: In a folder created by add_feature called feature1 which does not have a file called .pm_tag in it. For example, in /home/pat/cw2/project/feature1

```
$ pm add_tag F1
```

Post: A new file called .pm_tag exists in current directory, whose contents are the tag applied. e.g. cat .pm_tag should return: F1

3.9 </blockquote>

Pre: In a folder created by add_feature called feature1 which does have a file called .pm_tag in it. For example, in /home/pat/cw2/project/feature1/.pm_tag

```
$ pm add_tag F4
Already a tag for this folder
F1
```

Post: A new file called .pm_tag exists in current directory, whose contents are the tag applied. e.g. cat .pm_tag should return: F1

3.10 </blockquote>

3.10.1 find_tag

Pre: In a folder which has subfolders, one, two and three. Subfolder one has subfolders first and last. Folder one has a .pm_tag with contents F1, folder three has a .pm_tag with contents F3, and one/first has a .pm_tag with contents F1.1

```
$ pm find_tag F1
./one
```

Post: No change

Pre: In a folder which has subfolders, one, two and three. Subfolder one has subfolders first and last. Folder one has a .pm_tag with contents F1, folder three has a .pm_tag with contents F3, and one/first has a .pm_tag with contents F1.1

```
$ pm find_tag F1.1
./one/first
```

Post: No change

3.11 </blockquote>

3.11.1 move_by_tag

Pre: In a folder which has subfolders, one, two and three. Subfolder one has subfolders first and last. Folder one has a .pm_tag with contents F1, folder three has a .pm_tag with contents F3, and one/first has a .pm_tag with contents F1.1

```
$  pm move_by_tag F3 F1
```

Post: folder which relates to tag F3 (three in this case) should have moved from current folder (or whereever else it was...) to the folder related to F1 (one in this case)

3.12 </blockquote>

3.12.1 output_svg

Pre: In parent folder of project (for example in /home/pat/cw2 in this case).

```
$ pm output_svg myproject
```

Post: folder has a new file which is in plantuml wbs format, and a new .svg file created from it, and which does not list .git, bin, docs, lib, src or tests folders . e.g.

```
@startwbs
* myproject
** feature1
** one
*** first
*** last
** three
** two
@endwbs
```

Chapter 4

instructions

It is recommended to execute all of these commands in the topmost directory

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager PM
Opens the Project Manager, which is a text based user interface

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager CP
Gives a prompt for the user to input the name for the project, then creates a new project in the "Projects" folder

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager CF
Gives a prompt for the user to input the name for the feature and the project, then creates a new feature in the requested project directory

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager DP
Gives a prompt for the user to input the name of a project, then deletes the project

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager MP
Gives a prompt for the user to input the name of a project and the new location of the project, then moves the project

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager RP
Gives a prompt for the user to input the name of a project and the new name of the project, then renames the project

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager AT
Adds tags to all of the features in a project chosen by the user

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager AB
Gives a prompt for the user to input the name of the new branch and the project it goes in, then creates the branch

pc20_pm_cw2/ProjectManager/bin/LinkedProjectManager GT
Generates a text file containing every file in the file structure, which is used to generate a UMl diagram.

Chapter 5

Coursework 2 - a project manager tool

Using the file system as the main datastore, this tool helps manage work breakdown methodology, milestones and time management.

The core elements will be strongly guided, and completion of them equates to approximately 40% of the marks. Beyond that, students should problem solve, in a graded hierarchy of difficulty and completion to achieve higher marks.

It is unrealistic to expect to meet all the feature criteria listed

Part of the task therefore involves deciding which elements to implement in order to maximise your mark.

Some of the features may be best implemented using shell (bash or zsh) scripting, or could be done directly in C.

Tests will be provided for 1...10, which will also define the command line arguments required (including the sub-command name for the features, where appropriate e.g. pm feature feature_name or pm feature move new_↩ feature_name).

Coursework submission will include creating project documentation (using doxygen) so code ***must*** be well commented. Any shell scripts or make files included in the solution ***must*** be included in the documentation. The submission itself ***must*** be the URL of the CSGitLab repository used.

5.1 Core functionality:

5.1.1 Must (up to 10% for each top level feature)

1. Be able to create basic file structure for project
2. Abort if requested project/feature name already exists under 'root' folder. Here 'root' does not mean the / root of the file system, but the folder from which the program is run.
   (a) Requires checking existing file system for matching name
   (b) Requires using branching (*if*) to exit program if necessary
3. Initialise git repository
   (a) ***Should*** set up CSGitLab project etc.
4. Feature management
   (a) ***Must*** implement a method of having a shorthand code for feature e.g. F1, F2.1..., stored in a file.
   (b) ***Must*** implement lookup to facilitate getting path from shorthand code
   (c) ***Should*** include setting up git branch as appropriate

5.1.2 Should (up to 10% for each top level feature)

1. Include mechanism for renaming features (subtrees)
2. Include mechanism for moving feature to new location in tree (folder hierarchy)
3. Output tree diagram - PBS or WBS (svg, using plantuml)
   (a) Requires tree walk (iterative or recursive)
   (b) ***Must*** exclude folders that start with a '.'
   (c) ***Should*** use the plantuml tool
   (d) ***Could*** implement from scratch (much harder, more marks)
4. Time/workload estimate information stored in files in subfolders
   (a) ***Should*** have mechanisms for adding these from the program not just editing the files
   (b) ***Should*** include subtrees costs in parent tree total
5. Time/workload added to output diagram
   (a) ***Could*** also produce Gantt chart (using plantuml)

5.1.3 Could (up to 10% for each top level feature)

1. Output diagram includes links (when used in browser, for example)
   (a) ***Should*** use plantuml to do this
2. Dependencies information across tree branches
   (a) ***Must*** identify relevant other paths in tree to do this

5.1.4 Elite challenges ("Won't do" in MoSCoW terms) (up to 20% for each top level feature)

1. Guided work breakdown wizard (Slightly advanced, would require interactive questions/answer handling)
   (a) Needs a number of sub-features, such as minimum time allocation threshold, user input parsing
2. Multi-user (Advanced, would require some form of permissions model)
   (a) may be best done using a traditional SQL database, but can use flat files. Complex task.
3. Available as web application (Advanced, probably easiest creating a simple embedded server)
   (a) sample code for simple communications between applications will be covered
4. GOAP recommendation of suitable pathway (Advanced, can use existing GOAP library, however)
   (a) GOAP uses a 'heap' data structure

Chapter 6

File Index

6.1 File List

Here is a list of all files with brief descriptions:

Chapter 7

File Documentation

7.1 __FrontPage.md File Reference

7.2 DynamicLinkCompile.md File Reference

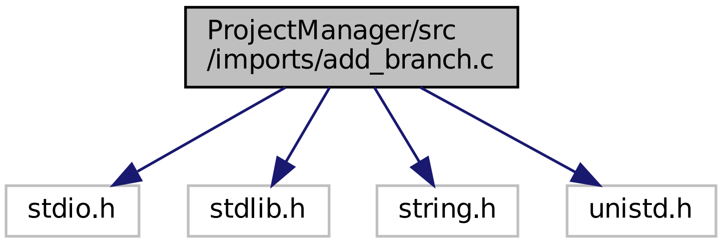
7.3 examples.md File Reference

7.4 instructions.md File Reference

7.5 ProjectManager/src/imports/add_branch.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

Include dependency graph for add_branch.c:



Functions

- int add_branch (char name[20])

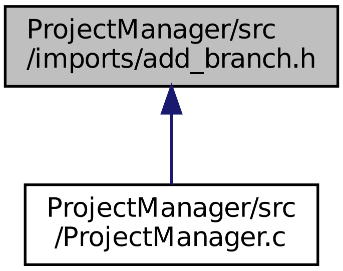
7.5.1 Function Documentation

7.5.1.1 add_branch()

```
int add_branch (
            char name[20] )
```

7.6 ProjectManager/src/imports/add_branch.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int add_branch (char name[20])

7.6.1 Function Documentation

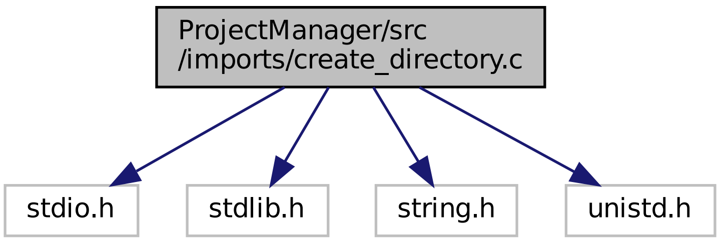
7.6.1.1 add_branch()

```
int add_branch (
            char name[20] )
```

7.7 ProjectManager/src/imports/create_directory.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
```

Include dependency graph for create_directory.c:



Functions

- int create_directory (char name[20])

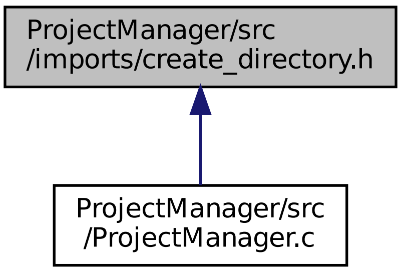
7.7.1 Function Documentation

7.7.1.1 create_directory()

```
int create_directory (
            char name[20] )
```

7.8 ProjectManager/src/imports/create_directory.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int create_directory (char name[20])

7.8.1 Function Documentation

7.8.1.1 create_directory()

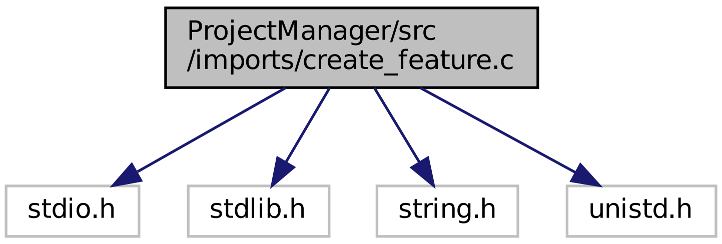
```
int create_directory (
            char name[20] )
```

7.9 ProjectManager/src/imports/create_feature.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#include <unistd.h>
```

Include dependency graph for create_feature.c:



Functions

- int create_feature (char name[20])

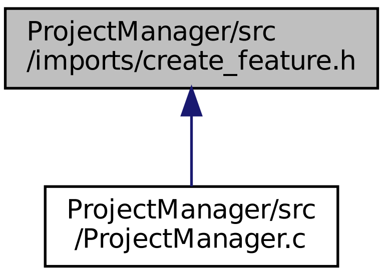
7.9.1 Function Documentation

7.9.1.1 create_feature()

```
int create_feature (
            char name[20] )
```

7.10 ProjectManager/src/imports/create_feature.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int create_feature (char name[20])

7.10.1 Function Documentation

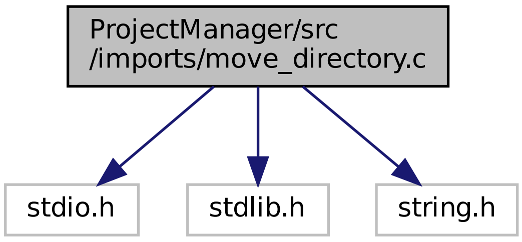
7.10.1.1 create_feature()

```
int create_feature (
            char name[20] )
```

7.11 ProjectManager/src/imports/move_directory.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for move_directory.c:



Functions

- int move_directory (char old_location[20], char new_location[20])

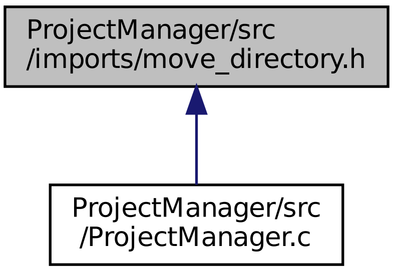
7.11.1 Function Documentation

7.11.1.1 move_directory()

```
int move_directory (
            char old_location[20],
            char new_location[20] )
```

7.12 ProjectManager/src/imports/move_directory.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int move_directory (char old_location[20], char new_location[20])

7.12.1 Function Documentation

7.12.1.1 move_directory()

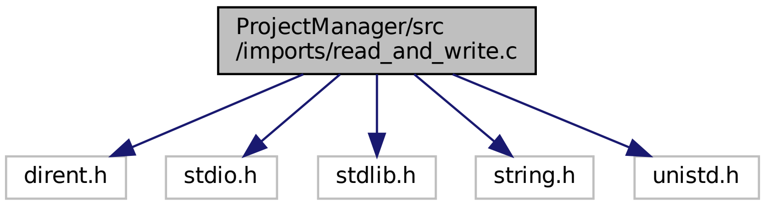
```
int move_directory (
            char old_location[20],
            char new_location[20] )
```

7.13 ProjectManager/src/imports/read_and_write.c File Reference

```
#include <dirent.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#include <unistd.h>
```

Include dependency graph for read_and_write.c:



Macros

- #define _DIR 1
- #define _FILE 2
- #define MAX_NUM 100
- #define NAME_LENGTH 100

Functions

- int write_tag (char tag[20], char title[20])
- int read_tag (char tag[20])
- void make_route (char result[ ], char path[ ], char fname[ ])
- int tag_tree_crawl (char type, char result[MAX_NUM][NAME_LENGTH], int index, char path[ ])
- int plantuml_tree_crawl (char type, char result[MAX_NUM][NAME_LENGTH], int index, char path[ ], char start[20])
- int create_tree (char name[20])
- int add_tags (void)
- int generate_tree (char start[20])

Variables

- const char ∗ dash ="/"

7.13.1 Macro Definition Documentation

7.13.1.1 _DIR

```
#define _DIR 1
```

7.13.1.2 _FILE

```
#define _FILE 2
```

7.13.1.3 MAX_NUM

```
#define MAX_NUM 100
```

7.13.1.4 NAME_LENGTH

```
#define NAME_LENGTH 100
```

7.13.2 Function Documentation

7.13.2.1 add_tags()

```
int add_tags (
             void  )
```

7.13.2.2 create_tree()

```
int create_tree (
             char name[20] )
```

7.13.2.3 generate_tree()

```
int generate_tree (
             char start[20] )
```

7.13.2.4 make_route()

```
void make_route (
            char result[],
            char path[],
            char fname[] )
```

7.13.2.5 plantuml_tree_crawl()

```
int plantuml_tree_crawl (
            char type,
            char result[MAX_NUM][NAME_LENGTH],
            int index,
            char path[],
            char start[20] )
```

7.13.2.6 read_tag()

```
int read_tag (
            char tag[20] )
```

7.13.2.7 tag_tree_crawl()

```
int tag_tree_crawl (
            char type,
            char result[MAX_NUM][NAME_LENGTH],
            int index,
            char path[] )
```

7.13.2.8 write_tag()

```
int write_tag (
            char tag[20],
            char title[20] )
```

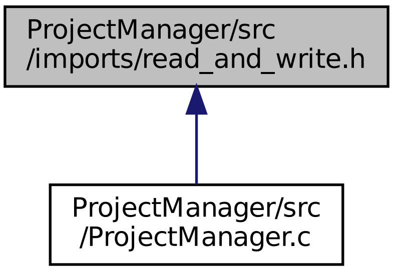
7.13.3 Variable Documentation

7.13.3.1 dash

```
const char* dash ="/"
```

7.14 ProjectManager/src/imports/read_and_write.h File Reference

This graph shows which files directly or indirectly include this file:


ProjectManager/src /imports/read_and_write.h
ProjectManager/src /ProjectManager.c

Macros

- #define _DIR 1
- #define _FILE 2
- #define MAX_NUM 100
- #define NAME_LENGTH 100

Functions

- void make_route (char result[ ], char path[ ], char fname[ ])
- int tag_tree_crawl (char type, char result[MAX_NUM][NAME_LENGTH], int index, char path[ ])
- int plantuml_tree_crawl (char type, char result[MAX_NUM][NAME_LENGTH], int index, char path[ ])
- int read_tag (char tag[20])
- int write_tag (char tag[20], char title[20])
- int add_tags (void)
- int generate_tree (char start[20])
- int create_tree (char name[20])

Variables

- const char ∗ dash ="/"

7.14.1 Macro Definition Documentation

7.14.1.1 _DIR

```
#define _DIR 1
```

7.14.1.2 _FILE

```
#define _FILE 2
```

7.14.1.3 MAX_NUM

```
#define MAX_NUM 100
```

7.14.1.4 NAME_LENGTH

```
#define NAME_LENGTH 100
```

7.14.2 Function Documentation

7.14.2.1 add_tags()

```
int add_tags (
             void  )
```

7.14.2.2 create_tree()

```
int create_tree (
             char name[20] )
```

7.14.2.3 generate_tree()

```
int generate_tree (
             char start[20] )
```

7.14.2.4 make_route()

```
void make_route (
            char result[],
            char path[],
            char fname[] )
```

7.14.2.5 plantuml_tree_crawl()

```
int plantuml_tree_crawl (
            char type,
            char result[MAX_NUM][NAME_LENGTH],
            int index,
            char path[] )
```

7.14.2.6 read_tag()

```
int read_tag (
            char tag[20] )
```

7.14.2.7 tag_tree_crawl()

```
int tag_tree_crawl (
            char type,
            char result[MAX_NUM][NAME_LENGTH],
            int index,
            char path[] )
```

7.14.2.8 write_tag()

```
int write_tag (
            char tag[20],
            char title[20] )
```

7.14.3 Variable Documentation

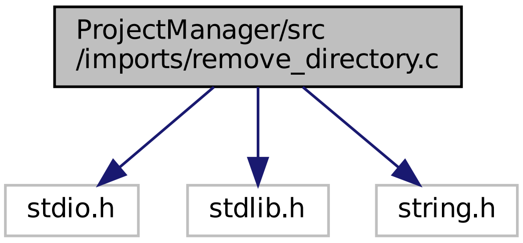
7.14.3.1 dash

```
const char* dash ="/"
```

7.15 ProjectManager/src/imports/remove_directory.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for remove_directory.c:



Functions

- int remove_directory (char name[20])

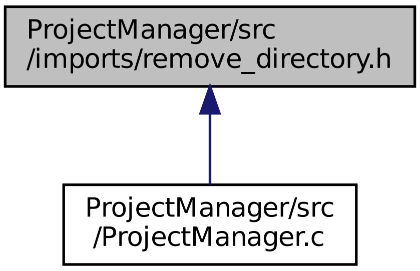
7.15.1 Function Documentation

7.15.1.1 remove_directory()

```
int remove_directory (
            char name[20] )
```

7.16 ProjectManager/src/imports/remove_directory.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int remove_directory (char name[20])

7.16.1 Function Documentation

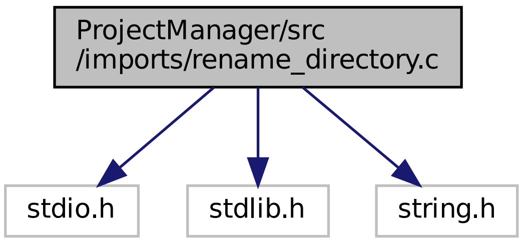
7.16.1.1 remove_directory()

```
int remove_directory (
            char name[20] )
```

7.17 ProjectManager/src/imports/rename_directory.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for rename_directory.c:



Functions

- int rename_directory (char old_name[20], char new_name[20])

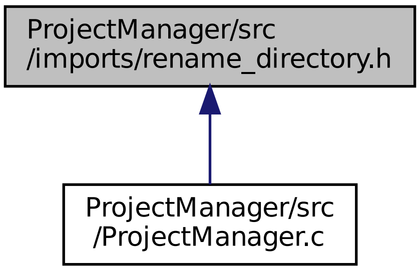
7.17.1 Function Documentation

7.17.1.1 rename_directory()

```
int rename_directory (
            char old_name[20],
            char new_name[20] )
```

7.18 ProjectManager/src/imports/rename_directory.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int rename_directory (char old_name[20], char new_name[20])

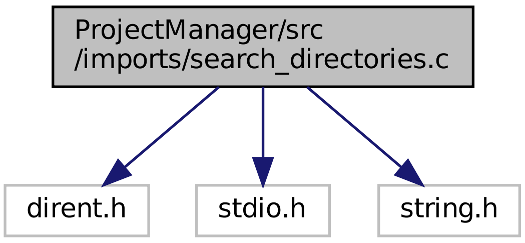
7.18.1 Function Documentation

7.18.1.1 rename_directory()

```
int rename_directory (
            char old_name[20],
            char new_name[20] )
```

7.19 ProjectManager/src/imports/search_directories.c File Reference

```
#include <dirent.h>
#include <stdio.h>
#include <string.h>
```
Include dependency graph for search_directories.c:



Macros

- #define _DIR 1
- #define _FILE 2
- #define MAX_NUM 100
- #define NAME_LENGTH 100

Functions

- void make_path (char result[ ], char path[ ], char fname[ ])
- int list_file_type (char type, char result[MAX_NUM][NAME_LENGTH], int index, char path[ ])
- int scan (void)
- int search_directory (char name[20])
- int search_files (char name[20])

Variables

- const char $*$ slash ="/"

7.19.1 Macro Definition Documentation

7.19.1.1 _DIR

```
#define _DIR 1
```

7.19.1.2 _FILE

```
#define _FILE 2
```

7.19.1.3 MAX_NUM

```
#define MAX_NUM 100
```

7.19.1.4 NAME_LENGTH

```
#define NAME_LENGTH 100
```

7.19.2 Function Documentation

7.19.2.1 list_file_type()

```
int list_file_type (
             char type,
             char result[MAX_NUM][NAME_LENGTH],
             int index,
             char path[] )
```

7.19.2.2 make_path()

```
void make_path (
             char result[],
             char path[],
             char fname[] )
```

7.19.2.3 scan()

```
int scan (
             void  )
```

7.19.2.4 search_directory()

```
int search_directory (
             char name[20] )
```

7.19.2.5 search_files()

```
int search_files (
             char name[20] )
```

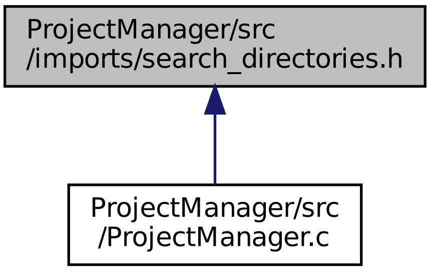
7.19.3 Variable Documentation

7.19.3.1 slash

```
const char* slash ="/"
```

7.20 ProjectManager/src/imports/search_directories.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define _DIR 1
- #define _FILE 2
- #define MAX_NUM 100
- #define NAME_LENGTH 100

Functions

- void make_path (char result[ ], char path[ ], char fname[ ])
- int list_file_type (char type, char result[MAX_NUM][NAME_LENGTH], int index, char path[ ])
- int scan (void)
- int search_directory (char name[20])
- int search_files (char name[20])

Variables

- const char ∗ slash ="/"
- int created

7.20.1 Macro Definition Documentation

7.20.1.1 _DIR

```
#define _DIR 1
```

7.20.1.2 _FILE

```
#define _FILE 2
```

7.20.1.3 MAX_NUM

```
#define MAX_NUM 100
```

7.20.1.4 NAME_LENGTH

```
#define NAME_LENGTH 100
```

7.20.2 Function Documentation

7.20.2.1 list_file_type()

```
int list_file_type (
             char type,
             char result[MAX_NUM][NAME_LENGTH],
             int index,
             char path[] )
```

7.20.2.2 make_path()

```
void make_path (
             char result[],
             char path[],
             char fname[] )
```

7.20.2.3 scan()

```
int scan (
             void  )
```

7.20.2.4 search_directory()

```
int search_directory (
             char name[20] )
```

7.20.2.5 search_files()

```
int search_files (
             char name[20] )
```

7.20.3 Variable Documentation

7.20.3.1 created

```
int created
```

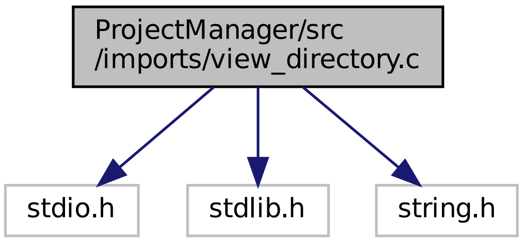
7.20.3.2 slash

```
const char* slash ="/"
```

7.21 ProjectManager/src/imports/view_directory.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

Include dependency graph for view_directory.c:



Functions

- int view_directory (void)

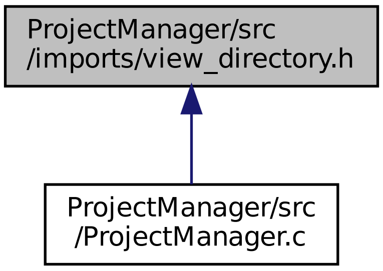
7.21.1 Function Documentation

7.21.1.1 view_directory()

```
int view_directory (
            void  )
```

7.22 ProjectManager/src/imports/view_directory.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int view_directory (void)

7.22.1 Function Documentation

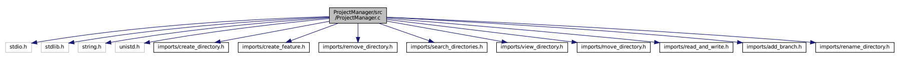
7.22.1.1 view_directory()

```
int view_directory (
             void  )
```

7.23 ProjectManager/src/ProjectManager.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include "imports/create_directory.h"
#include "imports/create_feature.h"
#include "imports/remove_directory.h"
#include "imports/search_directories.h"
#include "imports/view_directory.h"
#include "imports/move_directory.h"
#include "imports/read_and_write.h"
#include "imports/add_branch.h"
#include "imports/rename_directory.h"
```

Include dependency graph for ProjectManager.c:



Functions

- int ProjectManager (void)
- int main (int argc, const char **argv)

7.23.1 Function Documentation

7.23.1.1 main()

```
int main (
             int argc,
             const char ** argv )
```

7.23.1.2 ProjectManager()

```
int ProjectManager (
             void  )
```

7.24 README.md File Reference

Index