

TP1_b

Generated by Doxygen 1.8.13

Contents

1	Data Structure Index	1
1.1	Data Structures	1
2	File Index	3
2.1	File List	3
3	Data Structure Documentation	5
3.1	TParamDef Struct Reference	5
3.1.1	Detailed Description	5
3.1.2	Field Documentation	6
3.1.2.1	lettre	6
3.1.2.2	nom	6
3.1.2.3	type	6
3.1.2.4	valeur	6
3.2	TParamValue Union Reference	6
3.2.1	Detailed Description	7
3.2.2	Field Documentation	7
3.2.2.1	chaine	7
3.2.2.2	entier	7
3.2.2.3	reel	7

4 File Documentation	9
4.1 main.c File Reference	9
4.1.1 Function Documentation	9
4.1.1.1 main()	9
4.2 paramcmdl.c File Reference	10
4.2.1 Function Documentation	10
4.2.1.1 PrintParam()	11
4.2.1.2 ReadParamFromCommandLine()	11
4.2.1.3 ValeurParamToString()	11
4.2.2 Variable Documentation	12
4.2.2.1 ParamTypeChaine	12
4.3 paramcmdl.h File Reference	12
4.3.1 Enumeration Type Documentation	13
4.3.1.1 TParamType	13
4.3.2 Function Documentation	14
4.3.2.1 PrintParam()	14
4.3.2.2 ReadParamFromCommandLine()	14
4.3.2.3 ValeurParamToString()	15
4.3.3 Variable Documentation	15
4.3.3.1 ParamTypeChaine	15
Index	17

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

TParamDef	Represente un parametre de la ligne de commande (nom,type,valeur...)	5
TParamValue	Union permettant de manipuler un entier/reel/chaine	6

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

main.c	9
paramcmdl.c	10
paramcmdl.h	12

Chapter 3

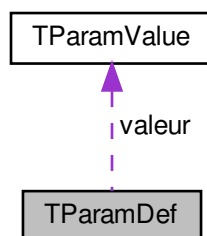
Data Structure Documentation

3.1 TParamDef Struct Reference

represente un parametre de la ligne de commande (nom,type,valeur...)

```
#include <paramcmdl.h>
```

Collaboration diagram for TParamDef:



Data Fields

- char * [nom](#)
- [TParamType](#) type
- char [lettre](#)
- [TParamValue](#) valeur

3.1.1 Detailed Description

represente un parametre de la ligne de commande (nom,type,valeur...)

3.1.2 Field Documentation

3.1.2.1 lettre

```
char TParamDef::lettre
```

lettre a utiliser sur la ligne de commande

3.1.2.2 nom

```
char* TParamDef::nom
```

nom du parametre

3.1.2.3 type

```
TParamType TParamDef::type
```

type (entier,reel,chaine)

3.1.2.4 valeur

```
TParamValue TParamDef::valeur
```

valeur a affecter au parametre

The documentation for this struct was generated from the following file:

- [paramcmdl.h](#)

3.2 TParamValue Union Reference

union permettant de manipuler un entier/reel/chaine

```
#include <paramcmdl.h>
```

Data Fields

- int [entier](#)
- float [reel](#)
- const char * [chaine](#)

3.2.1 Detailed Description

union permettant de manipuler un entier/reel/chaine

3.2.2 Field Documentation

3.2.2.1 chaine

```
const char* TParamValue::chaine
```

3.2.2.2 entier

```
int TParamValue::entier
```

3.2.2.3 reel

```
float TParamValue::reel
```

The documentation for this union was generated from the following file:

- [paramcmdl.h](#)

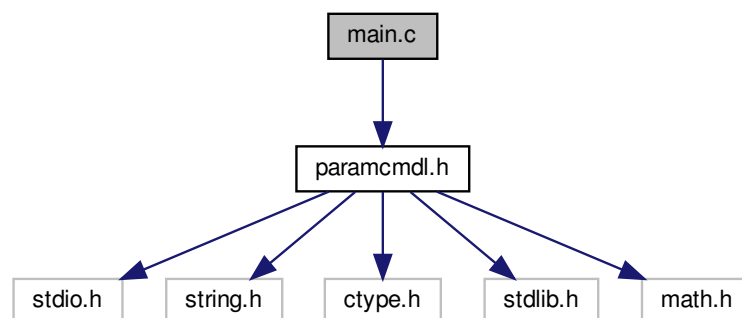
Chapter 4

File Documentation

4.1 main.c File Reference

```
#include "paramcmdl.h"
```

Include dependency graph for main.c:



Functions

- int `main` (const int `_argc`, const char *`_argv[]`)
fonction principale

4.1.1 Function Documentation

4.1.1.1 main()

```
int main (  
    const int _argc,  
    const char * _argv[] )
```

fonction principale

Parameters

in	<code>_argc</code>	: nombre d'arguments passes sur la ligne de commande
in	<code>_argv</code>	: tableau qui contient les chaines de caracteres passes en arguments du programme

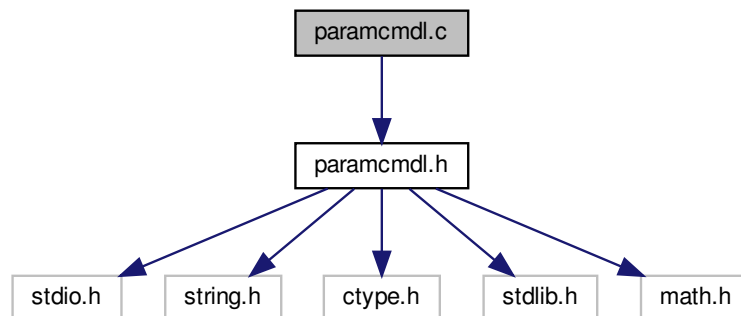
Returns

0 si terminaison normale

4.2 paramcmdl.c File Reference

```
#include "paramcmdl.h"
```

Include dependency graph for paramcmdl.c:

**Functions**

- char * [ValeurParamToString](#) (TParamDef * _tabParam, const int _index)
fonction qui transforme la valeur d'un parametre en chaine de caractere
- void [PrintParam](#) (TParamDef * _tabParam, const int _nbParam)
fonction qui affiche a l'ecran les parametre, leur type et leur valeur
- int [ReadParamFromCommandLine](#) (TParamDef * _tabParam, const int _nbParam, const int _argc, const char * _argv[])
fonction qui analyse la ligne de commande pour en extraire des valeurs pour les parametres

Variables

- char * [ParamTypeChaine](#) [] = {"entier", "reel", "chaine"}

4.2.1 Function Documentation

4.2.1.1 PrintParam()

```
PrintParam (
    TParamDef * _tabParam,
    const int _nbParam )
```

fonction qui affiche a l'ecran les parametre, leur type et leur valeur

Parameters

in	<code>_tabParam</code>	tableau des parametres de la ligne de commande
in	<code>_nbParam</code>	taille du tableau

Returns

neant

4.2.1.2 ReadParamFromCommandLine()

```
int ReadParamFromCommandLine (
    TParamDef * _tabParam,
    const int _nbParam,
    const int _argc,
    const char * _argv[ ] )
```

fonction qui analyse la ligne de commande pour en extraire des valeurs pour les parametres

Parameters

out	<code>_tabParam</code>	tableau des parametres de la ligne de commande
in	<code>_nbParam</code>	taille du tableau
in	<code>_argc</code>	nombre d'arguments passes sur la ligne de commande
in	<code>_argv</code>	tableau qui contient les chaines de caracteres passees en arguments du programme

Returns

≥ 0 : nombre de parametres mis a jour, < 0 : erreur

4.2.1.3 ValeurParamToString()

```
char * ValeurParamToString (
    TParamDef * _tabParam,
    const int _index )
```

fonction qui transforme la valeur d'un parametre en chaine de caractere

Parameters

in	<i>_tabParam</i>	tableau des parametres de la ligne de commande
in	<i>_index</i>	indice du parametre a considerer dans le tableau

Returns

une nouvelle chaine (qu'il faudra libérer par la suite)

4.2.2 Variable Documentation**4.2.2.1 ParamTypeChaine**

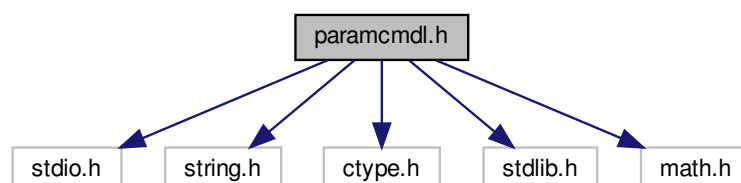
```
char* ParamTypeChaine[] = {"entier", "reel", "chaine"}
```

constante chaine de caracteres pour l'affichage des types

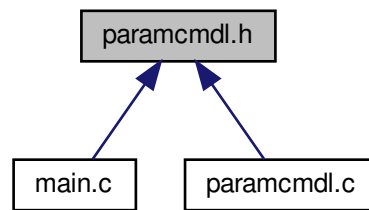
4.3 paramcmdl.h File Reference

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

Include dependency graph for paramcmdl.h:



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



Data Structures

- union [TParamValue](#)
union permettant de manipuler un entier/reel/chaine
- struct [TParamDef](#)
represente un parametre de la ligne de commande (nom,type,valeur...)

Enumerations

- enum [TParamType](#) { [PTentier](#), [PTreel](#), [PTchaine](#) }
Constantes pour le type des parametres de la ligne de commande.

Functions

- char * [ValeurParamToString](#) ([TParamDef](#) *_tabParam, const int _index)
fonction qui transforme la valeur d'un parametre en chaine de caractere
- void [PrintParam](#) ([TParamDef](#) *_tabParam, const int _nbParam)
fonction qui affiche a l'ecran les parametre, leur type et leur valeur
- int [ReadParamFromCommandLine](#) ([TParamDef](#) *_tabParam, const int _nbParam, const int _argc, const char *_argv[])
fonction qui analyse la ligne de commande pour en extraire des valeurs pour les parametres

Variables

- char * [ParamTypeChaine](#) []

4.3.1 Enumeration Type Documentation

4.3.1.1 TParamType

enum [TParamType](#)

Constantes pour le type des parametres de la ligne de commande.

Enumerator

PTentier	un nombre entier
PTreel	un nombre reel
PTchaîne	une chaîne de caracteres

4.3.2 Function Documentation**4.3.2.1 PrintParam()**

```
void PrintParam (
    TParamDef * _tabParam,
    const int _nbParam )
```

fonction qui affiche a l'ecran les parametre, leur type et leur valeur

Parameters

in	<i>_tabParam</i>	tableau des parametres de la ligne de commande
in	<i>_nbParam</i>	taille du tableau

Returns

neant

4.3.2.2 ReadParamFromCommandLine()

```
int ReadParamFromCommandLine (
    TParamDef * _tabParam,
    const int _nbParam,
    const int _argc,
    const char * _argv[] )
```

fonction qui analyse la ligne de commande pour en extraire des valeurs pour les parametres

Parameters

out	<i>_tabParam</i>	tableau des parametres de la ligne de commande
in	<i>_nbParam</i>	taille du tableau
in	<i>_argc</i>	nombre d'arguments passes sur la ligne de commande
in	<i>_argv</i>	tableau qui contient les chaînes de caracteres passees en arguments du programme

Returns

≥ 0 : nombre de parametres mis a jour, < 0 : erreur

4.3.2.3 ValeurParamToString()

```
char* ValeurParamToString (
    TParamDef * _tabParam,
    const int _index )
```

fonction qui transforme la valeur d'un parametre en chaine de caractere

Parameters

in	<i>_tabParam</i>	tableau des parametres de la ligne de commande
in	<i>_index</i>	indice du parametre a considerer dans le tableau

Returns

une nouvelle chaine (qu'il faudra libérer par la suite)

4.3.3 Variable Documentation

4.3.3.1 ParamTypeChaine

```
char* ParamTypeChaine[ ]
```

constante chaine de caracteres pour l'affichage des types

Index

- chaîne
 - [TParamValue, 7](#)
- entier
 - [TParamValue, 7](#)
- lettre
 - [TParamDef, 6](#)
- main
 - [main.c, 9](#)
- [main.c, 9](#)
 - [main, 9](#)
- nom
 - [TParamDef, 6](#)
- ParamTypeChaine
 - [paramcmdl.c, 12](#)
 - [paramcmdl.h, 15](#)
- [paramcmdl.c, 10](#)
 - [ParamTypeChaine, 12](#)
 - [PrintParam, 10](#)
 - [ReadParamFromCommandLine, 11](#)
 - [ValeurParamToString, 11](#)
- [paramcmdl.h, 12](#)
 - [ParamTypeChaine, 15](#)
 - [PrintParam, 14](#)
 - [ReadParamFromCommandLine, 14](#)
 - [TParamType, 13](#)
 - [ValeurParamToString, 15](#)
- PrintParam
 - [paramcmdl.c, 10](#)
 - [paramcmdl.h, 14](#)
- ReadParamFromCommandLine
 - [paramcmdl.c, 11](#)
 - [paramcmdl.h, 14](#)
- reel
 - [TParamValue, 7](#)
- [TParamDef, 5](#)
 - [lettre, 6](#)
 - [nom, 6](#)
 - [type, 6](#)
 - [valeur, 6](#)
- TParamType
 - [paramcmdl.h, 13](#)
- [TParamValue, 6](#)
 - [chaîne, 7](#)
 - [entier, 7](#)
- reel, [7](#)
- type
 - [TParamDef, 6](#)
- valeur
 - [TParamDef, 6](#)
- ValeurParamToString
 - [paramcmdl.c, 11](#)
 - [paramcmdl.h, 15](#)