

TP1\_b

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	TParamDef Struct Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Field Documentation . . . . .	5
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 . . . . .	6
3.2.2	Field Documentation . . . . .	6
3.2.2.1	chaine . . . . .	7
3.2.2.2	entier . . . . .	7
3.2.2.3	reel . . . . .	7
<b>4</b>	<b>File Documentation</b>	<b>9</b>
4.1	main.c File Reference . . . . .	9
4.1.1	Enumeration Type Documentation . . . . .	10
4.1.1.1	TParamType . . . . .	10
4.1.2	Function Documentation . . . . .	10
4.1.2.1	main() . . . . .	10
4.1.2.2	PrintParam() . . . . .	11
4.1.2.3	ReadParamFromCommandLine() . . . . .	11
4.1.2.4	ValeurParamToString() . . . . .	12
4.1.3	Variable Documentation . . . . .	12
4.1.3.1	ParamTypeChaine . . . . .	12
	<b>Index</b>	<b>13</b>



# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">TParamDef</a>	Represente un parametre de la ligne de commande (nom,type,valeur...) . . . . .	<a href="#">5</a>
<a href="#">TParamValue</a>	Union permettant de manipuler un entier/reel/chaine . . . . .	<a href="#">6</a>



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

<a href="#">main.c</a> . . . . .	9
----------------------------------	---





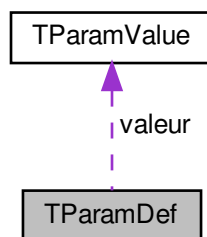
## Chapter 3

# Data Structure Documentation

### 3.1 TParamDef Struct Reference

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

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:

- [main.c](#)

## 3.2 TParamValue Union Reference

union permettant de manipuler un entier/reel/chaine

### 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:

- [main.c](#)



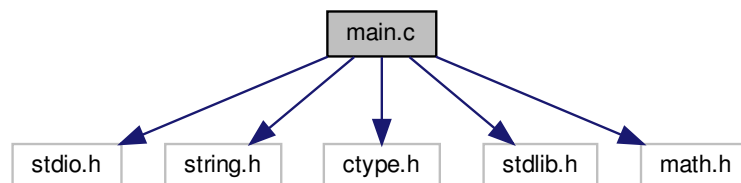
## Chapter 4

# File Documentation

### 4.1 main.c File Reference

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

Include dependency graph for main.c:



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

## Variables

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

### 4.1.1 Enumeration Type Documentation

#### 4.1.1.1 TParamType

enum [TParamType](#)

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

#### Enumerator

PTentier	un nombre entier
PTreel	un nombre reel
PTchaine	une chaine de caracteres

### 4.1.2 Function Documentation

#### 4.1.2.1 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

**4.1.2.2 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

**4.1.2.3 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

**4.1.2.4 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)

**4.1.3 Variable Documentation****4.1.3.1 ParamTypeChaine**

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

constante chaine de caracteres pour l'affichage des types



# Index

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