

LIBPQ wrapper c++

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	libpqwrp Namespace Reference . . . . .	7
4.1.1	Function Documentation . . . . .	7
4.1.1.1	HashStringToInt(const char *str, unsigned long long int hash=0) . . . . .	7
4.1.2	Variable Documentation . . . . .	8
4.1.2.1	delete0 . . . . .	8
4.1.2.2	fetch0 . . . . .	8
4.1.2.3	fetchEOF . . . . .	8
4.1.2.4	insert0 . . . . .	8
4.1.2.5	select0 . . . . .	8
4.1.2.6	update0 . . . . .	8

<b>5</b>	<b>Class Documentation</b>	<b>9</b>
5.1	libpqwrp::libPQwrp Class Reference	9
5.1.1	Detailed Description	11
5.1.2	Constructor & Destructor Documentation	11
5.1.2.1	libPQwrp()	11
5.1.2.2	~libPQwrp()	11
5.1.3	Member Function Documentation	11
5.1.3.1	begin()	11
5.1.3.2	cfield(int nfield)	12
5.1.3.3	clearRes()	12
5.1.3.4	closeDB()	12
5.1.3.5	commit()	12
5.1.3.6	connectDB(std::string info)	12
5.1.3.7	count_format_specifiers(std::string const &format)	12
5.1.3.8	countfield()	13
5.1.3.9	countrow()	13
5.1.3.10	DoubleToChar(double _X_, unsigned _precision_=0)	13
5.1.3.11	end()	13
5.1.3.12	fechsql(std::string cursor=""mycursor"")	13
5.1.3.13	fetch(int row, int column)	14
5.1.3.14	fetchAll(std::string, std::string cursor=""mycursor"")	14
5.1.3.15	fetchDbI(int row, int column)	14
5.1.3.16	fetchInt(int row, int column)	14
5.1.3.17	is_Table(const char *table)	14
5.1.3.18	nfield(std::string field)	15
5.1.3.19	opensql(std::string sql, std::string cursor=""mycursor"")	15
5.1.3.20	prepare(std::string const &format)	15
5.1.3.21	prepare(std::string const &format, Head &&head, Args &&...args)	15
5.1.3.22	qexec(std::string sql)	15
5.1.3.23	query(std::string sql, bool binary=true)	15

5.1.3.24	result()	16
5.1.3.25	rollback()	16
5.1.3.26	savpoint()	16
5.1.3.27	savpointRelease()	16
5.1.3.28	savpointRollback()	16
5.1.4	Friends And Related Function Documentation	16
5.1.4.1	operator>>	16
5.1.5	Member Data Documentation	17
5.1.5.1	clean	17
5.1.5.2	coln	17
5.1.5.3	cols	17
5.1.5.4	conn	17
5.1.5.5	fin_stage	17
5.1.5.6	first	17
5.1.5.7	FORMAT_SPECIFIER	17
5.1.5.8	res	17
5.1.5.9	rown	17
5.1.5.10	rows	18
5.1.5.11	sqlstream	18
<b>6</b>	<b>File Documentation</b>	<b>19</b>
6.1	libpqwrp.hpp File Reference	19
6.1.1	Macro Definition Documentation	20
6.1.1.1	DeLiMiTaTioN	20
6.1.1.2	NAMEOF	20
<b>Index</b>		<b>21</b>



# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">libpqwrp</a> . . . . .	7
------------------------------------	---





## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">libpqwrp::libPQwrp</a>	
Gestion parametre sql . . . . .	9



## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">libpqwrp.hpp</a> . . . . .	19
--	----



## Chapter 4

# Namespace Documentation

### 4.1 libpqwrp Namespace Reference

#### Classes

- class [libPQwrp](#)  
*gestion parametre sql*

#### Functions

- constexpr unsigned long long int [HashStringToInt](#) (const char \*str, unsigned long long int hash=0)  
*a function to make your alphanumeric switch*

#### Variables

- static bool [fetch0](#)
- static bool [update0](#)
- static bool [select0](#)
- static bool [delete0](#)
- static bool [insert0](#)
- static bool [fetchEOF](#)

#### 4.1.1 Function Documentation

4.1.1.1 constexpr unsigned long long int libpqwrp::HashStringToInt ( const char \* *str*, unsigned long long int *hash* = 0 )

a function to make your alphanumeric switch

Definition at line 70 of file libpqwrp.hpp.

## 4.1.2 Variable Documentation

### 4.1.2.1 `bool libpqwrp::delete0` `[static]`

Definition at line 62 of file `libpqwrp.hpp`.

### 4.1.2.2 `bool libpqwrp::fetch0` `[static]`

```
function sql de  
libpq
```

Definition at line 59 of file `libpqwrp.hpp`.

### 4.1.2.3 `bool libpqwrp::fetchEOF` `[static]`

Definition at line 64 of file `libpqwrp.hpp`.

### 4.1.2.4 `bool libpqwrp::insert0` `[static]`

Definition at line 63 of file `libpqwrp.hpp`.

### 4.1.2.5 `bool libpqwrp::select0` `[static]`

Definition at line 61 of file `libpqwrp.hpp`.

### 4.1.2.6 `bool libpqwrp::update0` `[static]`

Definition at line 60 of file `libpqwrp.hpp`.

## Chapter 5

# Class Documentation

### 5.1 libpqwrp::libPQwrp Class Reference

gestion parametre sql

```
#include <libpqwrp.hpp>
```

#### Public Member Functions

- [libPQwrp](#) ()
- [~libPQwrp](#) ()
- void [connectDB](#) (std::string info)  
*connect to the database*
- void [qexec](#) (std::string sql)  
*connect to the database*
- void [query](#) (std::string sql, bool binary=true)  
*PQexec.*
- char \* [fetch](#) (int row, int column)  
*PQquery.*
- double [fetchDbf](#) (int row, int column)  
*get the value of the row and the column*
- int [fetchInt](#) (int row, int column)  
*get the value of the row and the column*
- int [nfield](#) (std::string field)  
*get the value of the row and the column*
- const char \* [cfield](#) (int [nfield](#))  
*number column of the field*
- int [countfield](#) ()  
*name of the field*
- int [countrow](#) ()  
*number of columns*
- bool [is\\_Table](#) (const char \*table)  
*number of rows*
- void [closeDB](#) ()  
*if exist table of the database*

- void **begin** ()  
*close to the database*
- void **commit** ()  
*start transaction for commit*
- void **rollback** ()  
*commit for transaction init*
- void **end** ()  
*rollback for transaction init*
- void **clearRes** ()  
*end transaction for init*
- void **savpoint** ()  
*clear PGresult*
- void **savpointRollback** ()  
*point de sauvegarde*
- void **savpointRelease** ()  
*rollback base savepoint*
- void **fetchAll** (std::string, std::string cursor="mycursor")  
*delete savepoint*
- void **opensql** (std::string sql, std::string cursor="mycursor")  
*fetch ALL include requete*
- void **fetchsql** (std::string cursor="mycursor")  
*query for fetchsql record / record*
- template<typename Head , typename... Args>  
std::string **prepare** (std::string const &format, Head &&head, Args &&...args)  
*fetch record use openSQL*
- std::stringstream **result** ()  
*formatting the query with parameters*
- char \* **DoubleToChar** (double \_X\_, unsigned \_precision\_=0)  
*out buffer;*

## Public Attributes

- int **rows**  
*format without parameters causes an error*
- int **cols**  
*initialise par les requetes*
- int **rown**  
*initialise par les requetes*
- int **coln**  
*variable de traitement*
- PGresult \* **res**  
*variable de traitement*

## Protected Member Functions

- unsigned **count\_format\_specifiers** (std::string const &format)  
*contrôle PGresult*
- void **prepare** (std::string const &format)  
*count the number of parameters in the format*



- PGconn \* conn
- std::stringstream sqlstream
- bool first = true
- std::string fin\_stage
- bool clean

- static constexpr char **FORMAT SPECIFIER** = '?'

- `std::istream & operator>> (std::istream &is, char *)`  
*Double to char.*

## Definition at line 90 of file libpqwrp.hpp.

Definition at line 478 of file libpqwrp.hpp.

### 5.1.3.2 `const char * libpqwrp::libPQwrp::cfield ( int nfield )`

number column of the field

name of the field

Definition at line 422 of file libpqwrp.hpp.

### 5.1.3.3 `void libpqwrp::libPQwrp::clearRes ( )`

end transaction for init

clear PGresult

Definition at line 505 of file libpqwrp.hpp.

### 5.1.3.4 `void libpqwrp::libPQwrp::closeDB ( )`

if exist table of the database

close to the database

Definition at line 470 of file libpqwrp.hpp.

### 5.1.3.5 `void libpqwrp::libPQwrp::commit ( )`

start transction for commit

commit for transaction init

Definition at line 484 of file libpqwrp.hpp.

### 5.1.3.6 `void libpqwrp::libPQwrp::connectDB ( std::string info )`

connect to the database

init PGconn

init PGresult

Definition at line 312 of file libpqwrp.hpp.

### 5.1.3.7 `unsigned libpqwrp::libPQwrp::count_format_specifiers ( std::string const & format )` [protected]

contrôle PGresult

process

Definition at line 220 of file libpqwrp.hpp.

#### 5.1.3.8 int libpqwrp::libPQwrp::countfield ( )

name of the field

number of columns

Definition at line 434 of file libpqwrp.hpp.

#### 5.1.3.9 int libpqwrp::libPQwrp::countrow ( )

number of columns

number of rows

Definition at line 441 of file libpqwrp.hpp.

#### 5.1.3.10 char \* libpqwrp::libPQwrp::DoubleToChar ( double \_X\_, unsigned \_precision\_ = 0 )

out buffer;

Double to char.

Definition at line 660 of file libpqwrp.hpp.

#### 5.1.3.11 void libpqwrp::libPQwrp::end ( )

rollback for transaction init

end transaction for init

Definition at line 498 of file libpqwrp.hpp.

#### 5.1.3.12 void libpqwrp::libPQwrp::fetchsql ( std::string *cursor* = "mycursor" )

query for fetchsql record / record

fetch record use openSQL read a line record only

end of row or without row

close cursor

no end of row

Definition at line 631 of file libpqwrp.hpp.

#### 5.1.3.13 `char * libpqwrp::libPQwrp::fetch ( int row, int column )`

PQuery.

get the value of the row and the column

Definition at line 370 of file libpqwrp.hpp.

#### 5.1.3.14 `void libpqwrp::libPQwrp::fetchAll ( std::string sql, std::string cursor = "mycursor" )`

delete savepoint

fetch ALL include requete

———— process complex

lecture full rc ??? memory

end of row or without row

close cursor

no end of row

Definition at line 538 of file libpqwrp.hpp.

#### 5.1.3.15 `double libpqwrp::libPQwrp::fetchDbf ( int row, int column )`

get the value of the row and the column

Definition at line 383 of file libpqwrp.hpp.

#### 5.1.3.16 `int libpqwrp::libPQwrp::fetchInt ( int row, int column )`

get the value of the row and the column

Definition at line 396 of file libpqwrp.hpp.

#### 5.1.3.17 `bool libpqwrp::libPQwrp::is_Table ( const char * table )`

number of rows

if exist table of the database false for return value count

Definition at line 447 of file libpqwrp.hpp.

#### 5.1.3.18 int libpqwrp::libPQwrp::nfield ( std::string *field* )

get the value of the row and the column

number column of the field

Definition at line 409 of file libpqwrp.hpp.

#### 5.1.3.19 void libpqwrp::libPQwrp::opensql ( std::string *sql*, std::string *cursor* = "mycursor" )

fetch ALL include requete

query for fetchsql record / record end of row or without row

close cursor

no end of row

read a line record only

Definition at line 588 of file libpqwrp.hpp.

#### 5.1.3.20 void libpqwrp::libPQwrp::prepare ( std::string const & *format* ) [protected]

count the number of parameters in the format

Definition at line 239 of file libpqwrp.hpp.

#### 5.1.3.21 template<typename Head, typename... Args> std::string libpqwrp::libPQwrp::prepare ( std::string const & *format*, Head && *head*, Args &&... *args* )

fetch record use openSQL

formate la requete

Definition at line 248 of file libpqwrp.hpp.

#### 5.1.3.22 void libpqwrp::libPQwrp::qexec ( std::string *sql* )

connect to the database

PQexec.

Definition at line 328 of file libpqwrp.hpp.

#### 5.1.3.23 void libpqwrp::libPQwrp::query ( std::string *sql*, bool *binary* = true )

PQexec.

PQquery.

Definition at line 343 of file libpqwrp.hpp.

#### 5.1.3.24 `std::stringstream libpqwrp::libPQwrp::result ( )`

formatting the query with parameters

retrieve values on the fields

Definition at line 280 of file libpqwrp.hpp.

#### 5.1.3.25 `void libpqwrp::libPQwrp::rollback ( )`

commit for transaction init

rollback for transaction init

Definition at line 491 of file libpqwrp.hpp.

#### 5.1.3.26 `void libpqwrp::libPQwrp::savpoint ( )`

clear PGresult

point de sauvegarde

Definition at line 511 of file libpqwrp.hpp.

#### 5.1.3.27 `void libpqwrp::libPQwrp::savpointRelease ( )`

rollback base savepoint

delete savepoint

Definition at line 525 of file libpqwrp.hpp.

#### 5.1.3.28 `void libpqwrp::libPQwrp::savpointRollback ( )`

point de sauvegarde

rollback base savepoint

Definition at line 518 of file libpqwrp.hpp.

### 5.1.4 Friends And Related Function Documentation

#### 5.1.4.1 `std::istream& operator>> ( std::istream & is, char * t )` [friend]

Double to char.

retrieve char\* for stringstream dummy

Definition at line 197 of file libpqwrp.hpp.

### 5.1.5 Member Data Documentation

#### 5.1.5.1 `bool libpqwrp::libPQwrp::clean` `[private]`

Definition at line 105 of file libpqwrp.hpp.

#### 5.1.5.2 `int libpqwrp::libPQwrp::coln`

variable de traitement

Definition at line 124 of file libpqwrp.hpp.

#### 5.1.5.3 `int libpqwrp::libPQwrp::cols`

initialise par les requetes

Definition at line 121 of file libpqwrp.hpp.

#### 5.1.5.4 `PGconn* libpqwrp::libPQwrp::conn` `[private]`

Definition at line 94 of file libpqwrp.hpp.

#### 5.1.5.5 `std::string libpqwrp::libPQwrp::fin_stage` `[private]`

Definition at line 103 of file libpqwrp.hpp.

#### 5.1.5.6 `bool libpqwrp::libPQwrp::first = true` `[private]`

Definition at line 101 of file libpqwrp.hpp.

#### 5.1.5.7 `constexpr char libpqwrp::libPQwrp::FORMAT_SPECIFIER = '?'` `[static], [private]`

Definition at line 97 of file libpqwrp.hpp.

#### 5.1.5.8 `PGresult* libpqwrp::libPQwrp::res`

variable de traitement

Definition at line 127 of file libpqwrp.hpp.

#### 5.1.5.9 `int libpqwrp::libPQwrp::rown`

initialise par les requetes

Definition at line 123 of file libpqwrp.hpp.

#### 5.1.5.10 `int libpqwrp::libPQwrp::rows`

format without parameters causes an error

Definition at line 120 of file `libpqwrp.hpp`.

#### 5.1.5.11 `std::stringstream libpqwrp::libPQwrp::sqlstream` `[private]`

Definition at line 99 of file `libpqwrp.hpp`.

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

- [libpqwrp.hpp](#)



## Chapter 6

# File Documentation

### 6.1 libpqwrp.hpp File Reference

```
#include <string.h>
#include <cstdio>
#include <cstdlib>
#include <string>
#include <iostream>
#include <ostream>
#include <sstream>
#include <iomanip>
#include <stdexcept>
#include <libpq-fe.h>
#include <typeinfo>
```

#### Classes

- class [libpqwrp::libPQwrp](#)  
*gestion parametre sql*

#### Namespaces

- [libpqwrp](#)

#### Macros

- #define [NAMEOF](#)(variable) ((void)variable, #variable)
- #define [DeLiMiTaTioN](#) '~'

#### Functions

- constexpr unsigned long long int [libpqwrp::HashStringToInt](#) (const char \*str, unsigned long long int hash=0)  
*a function to make your alphanumeric switch*

## Variables

- static bool [libpqwrp::fetch0](#)
- static bool [libpqwrp::update0](#)
- static bool [libpqwrp::select0](#)
- static bool [libpqwrp::delete0](#)
- static bool [libpqwrp::insert0](#)
- static bool [libpqwrp::fetchEOF](#)

## 6.1.1 Macro Definition Documentation

### 6.1.1.1 `#define DeLiMiTaTioN '~'`

Definition at line 79 of file libpqwrp.hpp.

### 6.1.1.2 `#define NAMEOF( variable ) ((void)variable, #variable)`

Definition at line 76 of file libpqwrp.hpp.

# Index

- ~libPQwrp
  - libpqwrp::libPQwrp, [11](#)
- begin
  - libpqwrp::libPQwrp, [11](#)
- cfield
  - libpqwrp::libPQwrp, [11](#)
- clean
  - libpqwrp::libPQwrp, [17](#)
- clearRes
  - libpqwrp::libPQwrp, [12](#)
- closeDB
  - libpqwrp::libPQwrp, [12](#)
- coln
  - libpqwrp::libPQwrp, [17](#)
- cols
  - libpqwrp::libPQwrp, [17](#)
- commit
  - libpqwrp::libPQwrp, [12](#)
- conn
  - libpqwrp::libPQwrp, [17](#)
- connectDB
  - libpqwrp::libPQwrp, [12](#)
- count\_format\_specifiers
  - libpqwrp::libPQwrp, [12](#)
- countfield
  - libpqwrp::libPQwrp, [12](#)
- countrow
  - libpqwrp::libPQwrp, [13](#)
- DeLiMiTaTioN
  - libpqwrp.hpp, [20](#)
- delete0
  - libpqwrp, [8](#)
- DoubleToChar
  - libpqwrp::libPQwrp, [13](#)
- end
  - libpqwrp::libPQwrp, [13](#)
- FORMAT\_SPECIFIER
  - libpqwrp::libPQwrp, [17](#)
- fechsql
  - libpqwrp::libPQwrp, [13](#)
- fetch
  - libpqwrp::libPQwrp, [13](#)
- fetch0
  - libpqwrp, [8](#)
- fetchAll
  - libpqwrp::libPQwrp, [14](#)

- fetchDbI
  - libpqwrp::libPQwrp, [14](#)
- fetchEOF
  - libpqwrp, [8](#)
- fetchInt
  - libpqwrp::libPQwrp, [14](#)
- fin\_stage
  - libpqwrp::libPQwrp, [17](#)
- first
  - libpqwrp::libPQwrp, [17](#)
- HashStringToInt
  - libpqwrp, [7](#)
- insert0
  - libpqwrp, [8](#)
- is\_Table
  - libpqwrp::libPQwrp, [14](#)
- libPQwrp
  - libpqwrp::libPQwrp, [11](#)
- libpqwrp, [7](#)
  - delete0, [8](#)
  - fetch0, [8](#)
  - fetchEOF, [8](#)
  - HashStringToInt, [7](#)
  - insert0, [8](#)
  - select0, [8](#)
  - update0, [8](#)
- libpqwrp.hpp, [19](#)
  - DeLiMiTaTioN, [20](#)
  - NAMEOF, [20](#)
- libpqwrp::libPQwrp, [9](#)
  - ~libPQwrp, [11](#)
  - begin, [11](#)
  - cfield, [11](#)
  - clean, [17](#)
  - clearRes, [12](#)
  - closeDB, [12](#)
  - coln, [17](#)
  - cols, [17](#)
  - commit, [12](#)
  - conn, [17](#)
  - connectDB, [12](#)
  - count\_format\_specifiers, [12](#)
  - countfield, [12](#)
  - countrow, [13](#)
  - DoubleToChar, [13](#)
  - end, [13](#)
  - FORMAT\_SPECIFIER, [17](#)

- fechsql, [13](#)
  - fetch, [13](#)
  - fetchAll, [14](#)
  - fetchDbl, [14](#)
  - fetchInt, [14](#)
  - fin\_stage, [17](#)
  - first, [17](#)
  - is\_Table, [14](#)
  - libPQwrp, [11](#)
  - nfield, [14](#)
  - openssl, [15](#)
  - operator>>, [16](#)
  - prepare, [15](#)
  - qexec, [15](#)
  - query, [15](#)
  - res, [17](#)
  - result, [15](#)
  - rollback, [16](#)
  - rown, [17](#)
  - rows, [17](#)
  - savpoint, [16](#)
  - savpointRelease, [16](#)
  - savpointRollback, [16](#)
  - sqlstream, [18](#)
  - select0
    - libpqwrp, [8](#)
  - sqlstream
    - libpqwrp::libPQwrp, [18](#)
  - update0
    - libpqwrp, [8](#)
- NAMEOF
- libpqwrp.hpp, [20](#)
- nfield
- libpqwrp::libPQwrp, [14](#)
- openssl
- libpqwrp::libPQwrp, [15](#)
- operator>>
- libpqwrp::libPQwrp, [16](#)
- prepare
- libpqwrp::libPQwrp, [15](#)
- qexec
- libpqwrp::libPQwrp, [15](#)
- query
- libpqwrp::libPQwrp, [15](#)
- res
- libpqwrp::libPQwrp, [17](#)
- result
- libpqwrp::libPQwrp, [15](#)
- rollback
- libpqwrp::libPQwrp, [16](#)
- rown
- libpqwrp::libPQwrp, [17](#)
- rows
- libpqwrp::libPQwrp, [17](#)
- savpoint
- libpqwrp::libPQwrp, [16](#)
- savpointRelease
- libpqwrp::libPQwrp, [16](#)
- savpointRollback
- libpqwrp::libPQwrp, [16](#)