

Opt

phpPgWeb Reference Manual

Generated by Doxygen 1.2.18

Mon Nov 29 01:14:53 2004

Contents

1	phpPgWeb Main Page	1
2	phpPgWeb Compound Index	5
2.1	phpPgWeb Compound List	5
3	phpPgWeb File Index	7
3.1	phpPgWeb File List	7
4	phpPgWeb Class Documentation	9
4.1	Menu Class Reference	9
4.2	Pgdb Class Reference	13
4.3	Pgmessages Class Reference	19
4.4	Pgtable Class Reference	20
5	phpPgWeb File Documentation	37
5.1	config.ini File Reference	37
5.2	config.php File Reference	38
5.3	general.php File Reference	39
5.4	html.php File Reference	43
5.5	index.php File Reference	51
5.6	init.php File Reference	52
5.7	lobjview.php File Reference	53
5.8	main.dox File Reference	54
5.9	pgdb.php File Reference	55
5.10	pgmessage.php File Reference	56
5.11	pgtable.php File Reference	57
5.12	pgtbcnf.ini File Reference	58
5.13	postgres.php File Reference	59
5.14	retrocompatibilita.php File Reference	63
5.15	tables.php File Reference	65

Chapter 1

phpPgWeb Main Page

What this library can do

This **PHP** library is intended to make **quickly** a **user fiendly** web interface to a **postgresql** database. It is thought to be **configurable**, manage automatically **references betwhin tables**, has multilanguage support, and could be extended quite easily. It is not intended as admin tool (see **phpPgAdmin**), but as an intuitive web mask to a database. It supports also **file upload** that could be stored in your **file system** or directly in the database as a postgres **large object**. Creates **automatically thumbnail** if a **image** is uploaded and read **exif** information (metadata headers of image generated by digital cameras)

What requires

- It was tested on apache server on linux, but there are no reason to don't work with other systems.
- It requires PHP ≥ 4.0 but ≥ 4.2 it is better.

It is **important** that **php enviroment variable** `magic_quotes_gpc` is set to `Off`. **Set this in your php.ini file or in apache web server configuration file write** `php_admin_value magic_quotes_gpc Off`

- **It was tested with postgres ≥ 7.1 Postgres versions < 7.3 dont manage automatically relatuions betwin tables, but it is possible to link manually them.**
- **Thumbnail are made whit convert program of imagemagic if is installed, otherwise with php internal function if exist(if php is compiled with them).**
- **Exif info are read with php internal function if exist(if php is compiled with them).**

It works with all tested browser. With text browser links some non essential javascript function don't works, but it is usable.

How use this library

1. You must to include all `./include/libs` libraries (see `./include/init/init.php` for an example)
2. Than you have to open a postgresql connection with standard `pg_connect ()` function.
3. Now you can initialize a new **Pgtable** object, specifying as parameters postgresql connection object and configuration file (see `./conf/pgtbconf.ini` for an example). See **Pgtable Class Reference** for more details (or `./html/tables.php`).

4. So you have to add a new table to the object `Pgtable` with function `add_table()` specifying as parameters the schema name, the table name and if you want the primary key column name. See [Pgtable Class Reference](#) for more details (or [./html/tables.php](#)).
5. Now you can set some proprieties, configure visualization parameters, limit permission on the table and others features.
See [How upload files](#), [How change perms](#) or [how links tables](#) sections for examples.
6. Than you must to run the user query calling `switch_action()` method, specifying as parameters the schema name and the table name. This return a hash with keys:

- *text*: text to display at the bottom of the table
- *search_bar*: search bar
- *submit*: submit buttons
- *table*: html table

See [Pgtable Class Reference](#) for more details (or [./html/tables.php](#)).

7. So if you print `search_bar`, `table`, `text` and `submit` into a form tag that calls the same page, your script have to work. Note that the form must be a post thype form, with `enctype="multipart/form-data"` if you want submit files (see [./html/tables.php](#)). It could be similar to this:

```
<form name="pgform" method="POST" enctype="multipart/form-data" action="<?=$_SERVER["PHI
```

Note that it is important to specify form name that corespond with `formName` specified in the configuration file of your `Pgclass` object (see [./conf/pgtbconf.ini](#)). It is used by javascript for tables references.

How uplod files and store into the database

There are two mode to store files with this tool:

1. save as postgres **large object**. In this case files was stored directly into the database. This is the better way. There is a problem if database and php dont see the same filesystem. This could be resolved with a not implemented trigger procedure in plperl or other postgres procedural language.

Here is an example:

SQL code to create table:

```
CREATE TABLE lobjecttable(
note text,
file_size bigint,
file_name varchar,
file_type varchar,
file_thumbnail oid,
file_file oid,
file_exif varchar
);
```

now you have to call `alterType()` method of object `Pgtable` to change the visualization mode as specified at [point 5](#) of section [How use this library](#):


```

$pgtable->alterType('public', 'lobjtable','file_file', "lobj",
    array( "type"=>'',
        'mime_type_col'=>'file_type',
        'name_col'=>'file_name',
        'size_col'=>'file_size',
        'thumbnail_col'=>'file_thumbnail',
        'exif_col'=>'file_exif',
        "thumbnail_size"=>120
    )
);

```

”type” could be an array of mime_type allowed or of a part of mime_type allowed (for example "type"=>array("image", "application/pdf") if you want to store only images and pdf)

See **alterType()** function references for details or more features and [.html/tables.php](#) for an example.

2. save in a directory of the **filesystem** visible from web server. Into the database is saved only the file name. The directory is specified in the configuration file that you must specify when a new Pgclass object is initialized. See [./conf/pgtbconf.ini](#) for this detail.

Here is an example:

SQL code to create table:

```

CREATE TABLE filetable(
note text,
file_size bigint,
file_name varchar,
file_type varchar,
    file_thumbnail varchar,
file_file varchar,
file_exif varchar
);

```

now you have to call **alterType()** method of object [Pgtable](#) to change the visualization mode as specified at [point 5](#) of section [How use this library](#):

```

$pgtable->alterType('public', 'filetable','file_file', "file",
    array( "type"=>'',
        'mime_type_col'=>'file_type',
        'name_col'=>'file_name',
        'size_col'=>'file_size',
        'thumbnail_col'=>'file_thumbnail',
        'exif_col'=>'file_exif',
        "thumbnail_size"=>120
    )
);

```

”type” could be an array of mime_type allowed or of a part of mime_type allowed (for example "type"=>array("image", "application/pdf") if you want to store only images and pdf)

See **alterType()** function references for details or more features and [.html/tables.php](#) for an example.

For these two ways, there is a small problem of integrity: if your table in which there is stored the large object or the file name, could be modified from other interfaces different from phppgweb or has a reference on another table, created with "on delete cascade" clause, there is the possibility to remove only the record without deleting file or large object or the possibility to insert nonexistent files or large object. So only in these particular cases it is better to create a control trigger that delete large object or file when a delete occurs or when an update change the refer to the file.

How change language

Change language is simple: you have to modify the file specified by strings parameter into the configuration file of your Pgclass configuration file (specified when the class is initialized). (see [./conf/pgtbcconf.ini](#) for an example)

If your language not exist create a copy of existent file and translate the few strings presents in it. You can so send me this file to add support for other languages.

How link tables

Tables are automatically linked if are created with correctly postgresql syntax.

In other case you can join explicitly tables with **join_parent()** and **join_child()** methods of **Pgtable** class. If your referenced table is quite small and you want a select input against the default popup window, you could do this with **alter_parent()** method of **Pgtable** class. This is an example to be called as specified at [point 5](#) of section [How use this library](#):

```
$pgtable->alter_parent('child_table_schema_name', 'child_table_name', 'child_table_column_name
```

If you will display another column of referenced table or a mix of its column you can call the function in this way:

```
$pgtable->alter_parent('child_table_schema_name', 'child_table_name', 'child_table_column_name
                    array("type"=>"select",
                        "query"=>"SELECT ref_table_column as \"OID\", surname || ' ' || name as child_table_column
                        )
                    );
```

Note that it is important to rename appropriately the query results as in the example, and that OID correspond to referenced column name of child table.

The query must not have semicolon.

How manage perms

It is possible to change dynamically perms to a table (if view, insert, update...).

You have to redefine public attribute **actions** of **Pgtable** class.

By default it allows all actions on the selected tables table_name of schema table_schema_name: it is set to

```
$pgtable->actions[table_schema_name][table_name]=array('search','edit','insert','delete','view
```

You can reset this attribute to only some values of the above in the part specified at [point 5](#) of section [How use this library](#).

Chapter 2

phpPgWeb Compound Index

2.1 phpPgWeb Compound List

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

Menu (Class to manage menu)	9
Pgdb (This class is used to interface Pgtable class to postgresql,)	13
Pgmessages	19
Pgtable (The main class that is used to create the web interface to a postgresql database)	20

Chapter 3

phpPgWeb File Index

3.1 phpPgWeb File List

Here is a list of all files with brief descriptions:

config.ini	37
config.php	38
general.php	39
html.php	43
index.php	51
init.php	52
lobjview.php	53
main.dox	54
pgdb.php	55
pgmessage.php	56
pgtable.php	57
pgtbconf.ini	58
postgres.php	59
retrocompatibilita.php	63
tables.php	65

Chapter 4

phpPgWeb Class Documentation

4.1 Menu Class Reference

class to manage menu

Public Methods

- [Menu](#) ([\\$name](#)="Menu",[\\$attrib](#)="",[\\$type](#)="menu")
constructor
- [set_type](#) ([\\$type](#))
to set type of menu
- [add](#) ([\\$menu](#),[\\$position](#)="")
add a menu object into another menu object as submenu
- [rm](#) ([\\$position](#))
remove a submenu object from this menu
- [print_section](#) ([\\$position](#)="",[\\$print_selected](#),[\\$indent](#)=0)
return html code of submenu
- [print_menu](#) ([\\$position](#)="",[\\$print_all](#)=0,[\\$print_selected](#)=1,[\\$indent_px](#)=0,[\\$header](#)=1)
return menu html code

Public Attributes

- [type](#)
- [name](#)
- [sub_menu](#)
- [attrib](#)
- [position](#)

4.1.1 Detailed Description

class to manage menu

To create an html menu table

Parameters:

type= 'menu' css class

name name of menu section

sub_menu array of class Menu

attrib associative array of attributes of this section, these are inserted in <a> tag as key="value" as its attributes

position array of actual position

Definition at line 624 of file html.php.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Menu::Menu (\$ name = "Menu", \$ attrib = "", \$ type = "menu")

constructor

Parameters:

name= "Menu" name of the section

attrib= "" array of attributes for this section

type= 'default' not used

Definition at line 636 of file html.php.

References *attrib*, *name*, and *type*.

4.1.3 Member Function Documentation

4.1.3.1 Menu::add (\$ menu, \$ position = "")

add a menu object into another menu object as submenu

Parameters:

menu submenu

position= "" array of name of subsections in which insert submenu: if you have to add in root level, you must to set to "" or void array, if you have to add in subsection 'Level1' of subsection 'Level0' of root menu, \$position=array('Level0','Level1')

Definition at line 661 of file html.php.

References *position*, and *sub_menu*.

4.1.3.2 `Menu::print_menu` (*\$ position = ''*, *\$ print_all = 0*, *\$ print_selected = 1*, *\$ indent_px = 0*, *\$ header = 1*)

return menu html code

Parameters:

position= '' array of name of subsections in which you are: if you are in root level, you must to set to '' or void array, if you are in subsection 'Level1' of subsection 'Level0' of root menu, *\$position=array('Level0','Level1')*

print_all=0 if print all menu tree or only the "ramo" of *\$position*

print_selected=1 if print selected submenu

indent=0 pixel to indent submenus

header=1 if print <div> and tags

Definition at line 731 of file html.php.

References name, position, print_section(), sub_menu, and type.

4.1.3.3 `Menu::print_section` (*\$ position = ''*, *\$ print_selected*, *\$ indent = 0*)

return html code of submenu

Parameters:

position= '' name of current submenu

print_selected=1 if print selected submenu

indent=0 pixel to indent section

Definition at line 692 of file html.php.

References attrib, position, and sub_menu.

Referenced by print_menu().

4.1.3.4 `Menu::rm` (*\$ position*)

remove a submenu object from this menu

Parameters:

position= '' path of menu to remove: if you have to remove subsection 'Level1' in subsection 'Level0' of root menu, *\$position=array('Level0','Level1')*

Definition at line 676 of file html.php.

References position, and sub_menu.

4.1.3.5 `Menu::set_type` (*\$ type*)

to set type of menu

Parameters:

type type of menu

Definition at line 645 of file html.php.

References sub_menu, and type.

4.1.4 Member Data Documentation

4.1.4.1 Menu::**attrib**

Definition at line 628 of file html.php.

Referenced by Menu(), and print_section().

4.1.4.2 Menu::**name**

Definition at line 626 of file html.php.

Referenced by Menu(), and print_menu().

4.1.4.3 Menu::**position**

Definition at line 629 of file html.php.

Referenced by add(), print_menu(), print_section(), and rm().

4.1.4.4 Menu::**sub_menu**

Definition at line 627 of file html.php.

Referenced by add(), print_menu(), print_section(), rm(), and set_type().

4.1.4.5 Menu::**type**

Definition at line 625 of file html.php.

Referenced by Menu(), print_menu(), and set_type().

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

- [html.php](#)

4.2 Pgdb Class Reference

this class is used to interface [Pgtable](#) class to postgresql,

Public Methods

- [Pgdb](#) (\$dbConn)
- [check_pg_errors](#) ()
- [get_table_attributes](#) (\$tbSchema='public', \$tbName)
- [get_foreign_keys](#) (\$tbSchema='public', \$tbName)
- [get_inh_tables](#) (\$tbSchema='public', \$tbName)
- [set_value_from_pkey](#) (\$tbSchema='public', \$table, \$pkeyVal='', \$pkey='oid', \$query=false)
- [select](#) (\$query)
- [whereClause](#) (\$tbSchema, \$tbName, \$field, \$val, \$operator='=')
- [orderClause](#) (\$tbSchema, \$tbName, \$field, \$order='ASC')
- [limitClause](#) (\$limit=10, \$offset=0)
- [selectClause](#) (\$tbSchema, \$tbName, \$field='*', \$inherits)
- [numRecords](#) (\$tbSchema, \$tbName, \$inherits=true, \$where='')
- [delete](#) (\$tbSchema='public', \$tbName, \$pkey, \$pkeyField='oid', \$num=0)
- [update](#) (\$tbSchema='public', \$tbName, \$val, \$pkey, \$pkeyField='oid', \$num)
- [insert](#) (\$tbSchema='public', \$tbName, &\$val, \$pkeyfield, \$num=0)
- [not_null](#) (\$tbSchema, \$tbName)
- [is_expression](#) (\$tbSchema, \$tbName)
- [parse_geometry](#) (\$tbSchema, \$tbName, &\$val, \$field='')
- [filesEnd](#) ()
- [filesRollback](#) ()
- [newFileName](#) (\$oldName='')
- [checkUploadedFile](#) (\$schema, \$table, \$num, \$field)
- [fupload](#) (\$tbSchema, \$tbName, \$num, \$field, &\$arr)
- [fdelete](#) (\$tbSchema, \$tbName, \$pkeyField, \$pkeyVal, \$field, &\$arr)
- [readExif](#) (\$file)
- [clean_exif](#) (&\$arr)
- [export](#) (\$query)

Public Attributes

- [SQL_DEBUG](#)
- [dbConn](#)
- [pgVersion](#)
- [useSchema](#)
- [attribs](#)
- [notNull](#)
- [isExpression](#)
- [errors](#)
- [warnings](#)
- [files](#)
- [filesDir](#)
- [STRINGS](#)

4.2.1 Detailed Description

this class is used to interface [Pgtable](#) class to postgresql,

Definition at line 28 of file pgdb.php.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Pgdb::Pgdb (*\$ dbConn*)

Definition at line 50 of file pgdb.php.

References `dbConn`, `errors`, `files`, `filesDir`, `pgVersion`, `SQL_DEBUG`, `STRINGS`, `useSchema`, and `warnings`.

4.2.3 Member Function Documentation

4.2.3.1 Pgdb::check_pg_errors ()

Definition at line 73 of file pgdb.php.

4.2.3.2 Pgdb::checkUploadedFile (*\$ schema*, *\$ table*, *\$ num*, *\$ field*)

Definition at line 497 of file pgdb.php.

References `table`.

4.2.3.3 Pgdb::clean_exif (&*\$ arr*)

Definition at line 779 of file pgdb.php.

4.2.3.4 Pgdb::delete (*\$ tbSchema = 'public'*, *\$ tbName*, *\$ pkey*, *\$ pkeyField = 'oid'*, *\$ num = 0*)

Definition at line 248 of file pgdb.php.

References `pg_affected_rows()`, and `where_sql()`.

4.2.3.5 Pgdb::export (*\$ query*)

Definition at line 795 of file pgdb.php.

4.2.3.6 Pgdb::fdelete (*\$ tbSchema*, *\$ tbName*, *\$ pkeyField*, *\$ pkeyVal*, *\$ field*, &*\$ arr*)

Definition at line 633 of file pgdb.php.

4.2.3.7 Pgdb::filesEnd ()

Definition at line 462 of file pgdb.php.

4.2.3.8 Pgdb::filesRollback ()

Definition at line 473 of file pgdb.php.

4.2.3.9 Pgdb::fupload (\$ tbSchema, \$ tbName, \$ num, \$ field, &\$ arr)

Definition at line 543 of file pgdb.php.

4.2.3.10 Pgdb::get_foreign_keys (\$ tbSchema = 'public', \$ tbName)

Definition at line 109 of file pgdb.php.

References table.

4.2.3.11 Pgdb::get_inh_tables (\$ tbSchema = 'public', \$ tbName)

Definition at line 136 of file pgdb.php.

4.2.3.12 Pgdb::get_table_attributes (\$ tbSchema = 'public', \$ tbName)

Definition at line 81 of file pgdb.php.

4.2.3.13 Pgdb::insert (\$ tbSchema = 'public', \$ tbName, &\$ val, \$ pkeyfield, \$ num = 0)

Definition at line 343 of file pgdb.php.

References insert_sql(), and pg_affected_rows().

4.2.3.14 Pgdb::is_expression (\$ tbSchema, \$ tbName)

Definition at line 420 of file pgdb.php.

4.2.3.15 Pgdb::limitClause (\$ limit = 10, \$ offset = 0)

Definition at line 207 of file pgdb.php.

4.2.3.16 Pgdb::newFileName (\$ oldName = '')

Definition at line 485 of file pgdb.php.

4.2.3.17 Pgdb::not_null (\$ tbSchema, \$ tbName)

Definition at line 409 of file pgdb.php.

4.2.3.18 Pgdb::numRecords (\$ tbSchema, \$ tbName, \$ inherits = true, \$ where = '')

Definition at line 227 of file pgdb.php.

4.2.3.19 Pgdb::orderClause (*\$ tbSchema*, *\$ tbName*, *\$ field*, *\$ order = 'ASC'*)

Definition at line 199 of file pgdb.php.

4.2.3.20 Pgdb::parse_geometry (*\$ tbSchema*, *\$ tbName*, *&\$ val*, *\$ field = ' '*)

Definition at line 431 of file pgdb.php.

4.2.3.21 Pgdb::readExif (*\$ file*)

Definition at line 765 of file pgdb.php.

References `exif_read_data()`.

4.2.3.22 Pgdb::select (*\$ query*)

Definition at line 175 of file pgdb.php.

4.2.3.23 Pgdb::selectClause (*\$ tbSchema*, *\$ tbName*, *\$ field = "*"'*, *\$ inherits*)

Definition at line 216 of file pgdb.php.

4.2.3.24 Pgdb::set_value_from_pkey (*\$ tbSchema = 'public'*, *\$ table*, *\$ pkeyVal = ''*, *\$ pkey = 'oid'*, *\$ query = false*)

Definition at line 150 of file pgdb.php.

References `table`.

4.2.3.25 Pgdb::update (*\$ tbSchema = 'public'*, *\$ tbName*, *\$ val*, *\$ pkey*, *\$ pkeyField = 'oid'*, *\$ num*)

Definition at line 293 of file pgdb.php.

References `pg_affected_rows()`, `update_sql()`, and `where_sql()`.

4.2.3.26 Pgdb::whereClause (*\$ tbSchema*, *\$ tbName*, *\$ field*, *\$ val*, *\$ operator = '='*)

Definition at line 189 of file pgdb.php.

4.2.4 Member Data Documentation

4.2.4.1 Pgdb::attribs

Definition at line 35 of file pgdb.php.

4.2.4.2 Pgdb::dbConn

Definition at line 31 of file pgdb.php.

Referenced by Pgdb().

4.2.4.3 Pgdb::errors

Definition at line 39 of file pgdb.php.

Referenced by Pgdb().

4.2.4.4 Pgdb::files

Definition at line 42 of file pgdb.php.

Referenced by Pgdb().

4.2.4.5 Pgdb::filesDir

Definition at line 43 of file pgdb.php.

Referenced by Pgdb().

4.2.4.6 Pgdb::isExpression

Definition at line 37 of file pgdb.php.

4.2.4.7 Pgdb::notNull

Definition at line 36 of file pgdb.php.

4.2.4.8 Pgdb::pgVersion

Definition at line 32 of file pgdb.php.

Referenced by Pgdb().

4.2.4.9 Pgdb::SQL_DEBUG

Definition at line 29 of file pgdb.php.

Referenced by Pgdb().

4.2.4.10 Pgdb::STRINGS

Definition at line 45 of file pgdb.php.

Referenced by Pgdb().

4.2.4.11 Pgdb::useSchema

Definition at line 33 of file pgdb.php.

Referenced by Pgdb().

4.2.4.12 Pgdb::warnings

Definition at line 40 of file pgdb.php.

Referenced by Pgdb().

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

- [pgdb.php](#)

4.3 Pgmessages Class Reference

Public Methods

- [Pgmessages \(\)](#)
- [add \(\\$message=’'\)](#)
- [get \(\)](#)

Public Attributes

- [messages](#)

4.3.1 Constructor & Destructor Documentation

4.3.1.1 Pgmessages::Pgmessages ()

Definition at line 25 of file pgmessage.php.

References [messages](#).

4.3.2 Member Function Documentation

4.3.2.1 Pgmessages::add (\$ *message* = ’')

Definition at line 29 of file pgmessage.php.

References [messages](#).

4.3.2.2 Pgmessages::get ()

Definition at line 37 of file pgmessage.php.

References [messages](#).

4.3.3 Member Data Documentation

4.3.3.1 Pgmessages::messages

Definition at line 23 of file pgmessage.php.

Referenced by [add\(\)](#), [get\(\)](#), and [Pgmessages\(\)](#).

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

- [pgmessage.php](#)

4.4 Pgtable Class Reference

the main class that is used to create the web interface to a postgresql database

Public Methods

- **Pgtable** (\$db_conn,\$configFile,\$user="", \$group="", \$configSchema="")
initialize pgtable object
- **sort_field** ()
reorder the table, how \$indexPriority specify
- **set_priority** (\$priority=0,\$schema=false,\$table=false,\$field=false)
set the priority of a column and reorder columns
- **set_perms_priority** (\$tbSchema,\$tbName)
set default perms and priority number of a table
- **set_defaults** (\$tbSchema,\$tbName)
set defaults variable when a new table is initialized
- **add_table** (\$tbSchema,\$tbName,\$pkey=false)
initialize a new table
- **join_child** (\$schemaParent='public',\$parent,\$parentcol,\$schemaChild='public',\$child,\$childcol)
join an existent parent table with its childs, adding links down view/edit pages
- **join_parent** (\$schemaChild='public',\$child,\$childcol,\$schemaParent='public',\$parent,\$parentcol,\$schema='public',\$sql='view', 'preSearch', 'new', 'edit'))
join an existent child table with its parent,
- **alter_parent** (\$schemaChild,\$child,\$childcol,\$arr=array())
change parent references attributes. Used to alter default values of prelinked tables(for examples type or to specify a query)
- **auto_join** (\$tbSchema,\$tbName,\$field,\$when)
called when necessary from internal functions to set form defaults or similar
- **auto_join_query** (\$tbSchema,\$tbName,\$when)
called when necessary from internal functions to set query with joins
- **operator** (\$name,\$type="", \$defVal=false)
return appropriate html code for an operator
- **delete** (\$tbSchema,\$tbName)
called when the html button 'Delete' is clicked
- **update** (\$tbSchema,\$tbName)
called when the html button 'Save' is clicked (only save modified data)

- **copy** (\$tbSchema,\$tbName)
called when the html button 'Copy' is clicked
- **insert** (\$tbSchema,\$tbName)
called when the html button 'Insert' is clicked (only insert new data)
- **pre_search_new** (\$tbSchema,\$tbName)
called when the html button 'new search' is clicked
- **pre_search_modify** (\$tbSchema,\$tbName)
called when the html button 'modify search' is clicked
- **pre_search** (\$tbSchema,\$tbName,\$new=true)
called when from pre_search_modify and pre_search_new functions
- **view** (\$tbSchema,\$tbName)
called when the html button 'View' is clicked
- **edit** (\$tbSchema,\$tbName)
called when the html button 'edit' or 'insert' is clicked
- **pre_insert** (\$tbSchema,\$tbName)
called when the html button 'new insert' is clicked
- **view_edit** (\$tbSchema,\$tbName,\$when,&\$datain)
called from others functions
- **order** (\$tbSchema,\$tbName,\$field)
for order links in search table
- **actions** (\$tbSchema,\$tbName,\$field,\$num,\$pkey,\$refLink=array(),\$type="checkbox")
add links for each records to modify, delete, view and child subtables if exists
- **export** (\$tbSchema,\$tbName)
called when the html button 'export' is clicked, export current selected records in dbf
- **search** (\$tbSchema,\$tbName,\$export=false)
called when the html button 'search' is clicked
- **parse_data_in** (&\$arr,\$when="")
parse submitted data
- **parse_data_out** (\$when,\$tbSchema,\$tbName)
parse output data hash, from a bidimensional array of values(corespond to a part of postgres table), create a html table structured in a bidimensinal hash
- **form_sub** (\$tbSchema,\$tbName,\$field,\$num,\$when,\$def)
called for each field, to make an appropriate input form

- [view_sub](#) (\$tbSchema,\$tbName,\$field,\$num,\$when,\$def)
called for each field, to view its value in an appropriate parsed mode
- [alter_display](#) (\$schema,\$table,\$fields=false,\$value=true,\$when=array("new","edit","search","view","pre-Search","export"))
set when a column is displayed or not
- [alter_type](#) (\$tbSchema="public",\$tbName,\$tbField,\$type,\$properties=false,\$changePropriety=true,\$where=false)
change type of a field
- [file_url](#) (\$file,\$object=false,\$mimeType="")
- [file_view](#) (\$linkName,\$link,\$thumbnail,\$mimeType)
- [switch_actions](#) (\$tbSchema='public',\$tbName)
main function
- [switch_edit](#) (\$tbSchema,\$tbName)
- [get_table](#) ()
to be called to know how table is requested
- [search_bar](#) (\$searchUrl=false,\$previousUrl=false,\$nextUrl=false,\$otherUrl=false,\$total=false,\$offset=false,\$limit=false,\$numPages=false,\$nameOffset)
- [submit](#) (\$tbSchema,\$tbName,\$arr)

Public Attributes

- [db](#)
- [pkey](#)
- [ref](#)
- [inh](#)
- [perms](#)
- [priority](#)
- [form](#)
- [data](#)
- [links](#)
- [actions](#)
- [user](#)
- [group](#)
- [configSchema](#)
- [operator](#)
- [indexPriority](#)
- [errors](#)
- [messages](#)
- [STRINGS](#)

4.4.1 Detailed Description

the main class that is used to create the web interface to a postgresql database

Parameters:

db Pgdb object, to interface with postgresql db

pkey name of column to use as a pkey (\$pkey[schema][table]=column)

ref hash of references to other tables

inh if a table inherits other tables

perms not used, to manage perms in the future

priority table priority, not used

form html form informations

actions array of action allowed (edit, insert....)

operator array of operator for each pseudotype

user not used

group not used

configSchema not used

indexPriority priority priority of column, if must be reorder from postgres column order (\$priority[schema][table][column]=number)

erros Pgmessage object

messages Pgmessage object

STRINGS array of strings, form multilanguage support

Definition at line 45 of file pgtable.php.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 Pgtable::Pgtable (\$db_conn, \$configFile, \$user = "", \$group = "", \$configSchema = "")

initialize pgtable object

Parameters:

db_conn valid postgresql connection (the returned value of pg_connect() function)

configFile ini file in wich there are set some variables as icons urls etc.

users not used

group not used

configSchema not used

Returns:

Definition at line 83 of file pgtable.php.

References configSchema, data, db, errors, form, group, messages, perms, priority, STRINGS, and user.

4.4.3 Member Function Documentation

4.4.3.1 Pgtable::actions (*\$tbSchema*, *\$tbName*, *\$field*, *\$num*, *\$pkey*, *\$refLink* = array(), *\$type* = "checkbox")

add links for each records to modify, delete, view and child subtables if exists

Parameters:

tbSchema schema name of the table
tbName table name
field column name
num line number of current sets of records
pkey pkey value of current record
refLink hash to manage references to subtables
type= "checkbox" if use a checkbox or a hidden input type

Returns:

html code

Definition at line 801 of file pgtable.php.

References title.

4.4.3.2 Pgtable::add_table (*\$tbSchema*, *\$tbName*, *\$pkey* = false)

initialize a new table

Parameters:

tbSchema schema of table
tbName table name
pkey the primary key column of the table

Definition at line 254 of file pgtable.php.

References db, inh, pkey, ref, set_defaults(), and set_perms_priority().

4.4.3.3 Pgtable::alter_display (*\$schema*, *\$table*, *\$fields* = false, *\$value* = true, *\$when* = array("new", "edit", "search", "view", "preSearch", "export"))

set when a column is displayed or not

Note:

to call before alter_type if alter_type is called!!!

Parameters:

tbSchema schema name of the table
tbName table name
field column name, or arrays of column names, or false if all the columns
value=true if true view selected columns, if false hide they
when when apply current changes (new, edit, search...) it could be an array

Definition at line 1383 of file pgtable.php.

References table.

4.4.3.4 Pgtable::alter_parent (\$ schemaChild, \$ child, \$ childcol, \$ arr = array())

change parent references attributes. Used to alter default values of prelinked tables(for examples type or to specify a query)

Parameters:

schemaChild schema of child table

child child table

childcol child table column

arr hash with this facultative keys: table, field, query, type=>(popup or select), when=>array('search', 'view', 'preSearch', 'insert', 'edit')

Note:

\$query must is a valid sql query that returns two columns, first must be renamed (with AS syntax) as "OID", the second as \$field. First must correspond to values to insert in \$tbName.\$field and second as a label of this.

\$query must have no semicolon

Definition at line 323 of file pgtable.php.

References ref.

4.4.3.5 Pgtable::alter_type (\$ tbSchema = "public", \$ tbName, \$ tbField, \$ type, \$ proprieties = false, \$ change_propriety = true, \$ where = false)

change type of a field

Parameters:

tbSchema= 'public' schema name

tbName table name

tbField field name

type new type to change, eg "file" or "lobject" to store files directly into the database

proprieties : array for file or lobject types: it could be contain key as: maxsize (in byte) type (it could be an array, and is matched with mime-type) mime_type_col (column of the same table in wich send mime_type of the file) name_col (column of the same table in wich send filename) size_col (column of the same table in wich send file size in byte) thumbnail_col (column of the same table of type oid in wich send thumbnail) thumbnail.size (size in pixel of thumbnail) exif_col (column of the same table in wich send exif info of the image) mime_type (if you would insert a unique type of file, and so mime_type_col and type are not necessary)

change_propriety if alter display array

where=false if change only form or attribs array or both

Definition at line 1423 of file pgtable.php.

4.4.3.6 Pgtable::auto_join (\$ tbSchema, \$ tbName, \$ field, \$ when)

called when necessary from internal functions to set form defaults or similar

Parameters:

when when this function is called (search, insert, edit...)

Definition at line 334 of file phtable.php.

References db, form, pkey, ref, STRINGS, and table.

4.4.3.7 Pgtable::auto_join_query (\$ tbSchema, \$ tbName, \$ when)

called when necessary from internal functions to set query with joins

Parameters:

when when this function is called (search, insert, edit....)

Definition at line 367 of file phtable.php.

References db, form, pg_escape_string(), pkey, ref, search(), table, and view().

4.4.3.8 Pgtable::copy (\$ tbSchema, \$ tbName)

called when the html button 'Copy' is clicked

Parameters:

tbSchema schema name of the table

tbName table name

Returns:

bidimensional array, that correspond to html table

Definition at line 519 of file phtable.php.

Referenced by set_defaults().

4.4.3.9 Pgtable::delete (\$ tbSchema, \$ tbName)

called when the html button 'Delete' is clicked

Parameters:

tbSchema schema name of the table

tbName table name

Definition at line 454 of file phtable.php.

References array_unset().

4.4.3.10 Pgtable::edit (\$ tbSchema, \$ tbName)

called when the html button 'edit' or 'insert' is clicked

Parameters:

tbSchema schema name of the table

tbName table name

Returns:

bidimensional array, that correspond to html table

Definition at line 624 of file phtable.php.

Referenced by join_parent(), and set_defaults().

4.4.3.11 Pgtable::export (\$tbSchema, \$tbName)

called when the html button 'export' is clicked, export current selected records in dbf

Parameters:

tbSchema schema name of the table

tbName table name

Returns:

file link

Definition at line 862 of file pgtable.php.

4.4.3.12 Pgtable::file_url (\$file, \$lobject = false, \$ mimeType = "")

Definition at line 1479 of file pgtable.php.

4.4.3.13 Pgtable::file_view (\$linkName, \$link, \$thumbnail, \$ mimeType)

Definition at line 1492 of file pgtable.php.

4.4.3.14 Pgtable::form_sub (\$tbSchema, \$tbName, \$field, \$ num, \$ when, \$ def)

called for each field, to make an appropriate input form

Parameters:

tbSchema schema name of the table

tbName table name

field column name

num line number of current sets of records

when when is calles(search, edit, insert...)

def current value of the field

Definition at line 1209 of file pgtable.php.

4.4.3.15 Pgtable::get_table ()

to be called to know how table is requested

Returns:

list(schemaname, tablename, primary key)

Definition at line 1702 of file pgtable.php.

4.4.3.16 Pgtable::insert (\$ tbSchema, \$ tbName)

called when the html button 'Insert' is clicked (only insert new data)

Parameters:

tbSchema schema name of the table

tbName table name

Definition at line 530 of file pgtable.php.

Referenced by set_defaults().

4.4.3.17 Pgtable::join_child (\$ schemaParent = 'public', \$ parent, \$ parentcol, \$ schemaChild = 'public', \$ child, \$ childcol)

join an existent parent table with its childs, adding links down view/edit pages

Note:

if db is created with foeking key or references commandes, this is called automatically

Parameters:

schemaParent schema of parent table

parent parent table

parentcol parent table column

schemaChild schema of child table

child child table

childcol child table column

Definition at line 285 of file pgtable.php.

References ref, and table.

4.4.3.18 Pgtable::join_parent (\$ schemaChild = 'public', \$ child, \$ childcol, \$ schemaParent = 'public', \$ parent, \$ parentcol, \$ schema = 'public', \$ query = "", \$ type = "popup", \$ when = array('search', 'view', 'preSearch', 'new', 'edit'))

join an existent child table with its parent,

Note:

if db is created with foeking key or references commandes, this is called automatically and a popup window is opened when necessary. In this case, for small parent references, it is sometimes better use a select list of possible values: see [alter_parent\(\)](#) function for this.

Parameters:

schemaChild schema of child table

child child table

childcol child table column

schemaParent schema of parent table

parent parent table

parentcol parent table column

query valid sql query that select all valid childs

Note:

\$query must is a valid sql query that returns two columns, first must be renamed (with AS syntax) as "OID", the second as \$field. First must correspond to values to insert in \$tbName.\$field and second as a label of this.

\$query must have no semicolon

Parameters:

type= 'popup' or 'select' : the first open a new window, the second make a select menu(DONT USE THIS FOR LONG TABLES!!)

when=array ('search', 'view', 'preSearch', 'insert', 'edit') when join

Definition at line 307 of file pgtable.php.

References edit(), ref, search(), table, and view().

4.4.3.19 Pgtable::operator (\$ name, \$ type = "", \$ defVal = false)

return appropriate html code for an operator

Parameters:

name html input name value

type type of current field(int, text...)

defVal= "" default value

Definition at line 423 of file pgtable.php.

References html_select().

4.4.3.20 Pgtable::order (\$ tbSchema, \$ tbName, \$ field)

for order links in search table

Parameters:

tbSchema schema name of the table

tbName table name

field column name

Returns:

html code

Definition at line 771 of file pgtable.php.

References arr_to_url(), and title.

4.4.3.21 Pgtable::parse_data_in (&\$ arr, \$ when = '')

parse submitted data

Note:

because of recursion of query, the following functions must work also if is called more than once on the same data : so don't could for example add a constant string to a parameter because in some cases it adds more times the same string....

Parameters:

arr hash to parse

when when is called (search, edit, insert...)

Definition at line 1072 of file pgtable.php.

References table.

4.4.3.22 Pgtable::parse_data_out (\$ when, \$ tbSchema, \$ tbName)

parse output data hash, from a bidimensional array of values (correspond to a part of postgres table), create a html table structured in a bidimensional hash

Parameters:

when when is called (search, edit, insert...)

tbSchema schema of table

tbName table name

Definition at line 1142 of file pgtable.php.

4.4.3.23 Pgtable::pre_insert (\$ tbSchema, \$ tbName)

called when the html button 'new insert' is clicked

Parameters:

tbSchema schema name of the table

tbName table name

Returns:

bidimensional array, that correspond to html table

Definition at line 637 of file pgtable.php.

4.4.3.24 Pgtable::pre_search (\$ tbSchema, \$ tbName, \$ new = true)

called when from pre_search_modify and pre_search_new functions

Parameters:

tbSchema schema name of the table

tbName table name

new if is a new query

Returns:

bidimensional array, that corespodnd to html table

Definition at line 585 of file pgtable.php.

4.4.3.25 Pgtable::pre_search_modify (\$ tbSchema, \$ tbName)

called when the html button 'modify search' is clicked

Parameters:

tbSchema schema name of the table

tbName table name

Returns:

bidimensional array, that corespond to html table

Definition at line 572 of file pgtable.php.

4.4.3.26 Pgtable::pre_search_new (\$ tbSchema, \$ tbName)

called when the html button 'new search' is clicked

Parameters:

tbSchema schema name of the table

tbName table name

Returns:

bidimensional array, that corespond to html table

Definition at line 560 of file pgtable.php.

4.4.3.27 Pgtable::search (\$ tbSchema, \$ tbName, \$ export = false)

called when the html button 'search' is clicked

Parameters:

tbSchema schema name of the table

tbName table name

export=false

Returns:

bidimensional array corespond to html table if \$export=false, sql query if \$export=true

Definition at line 884 of file pgtable.php.

References array_unset(), getmicrotime(), transpose(), and where_sql().

Referenced by auto_join_query(), join_parent(), and set_defaults().

4.4.3.28 Pgtable::search_bar (*\$ searchUrl = false, \$ previousUrl = false, \$ nextUrl = false, \$ otherUrl = false, \$ total = false, \$ offset = false, \$ limit = false, \$ numPages = false, \$ nameOffset*)

Definition at line 1717 of file pgtable.php.

4.4.3.29 Pgtable::set_defaults (*\$ tbSchema, \$ tbName*)

set defaults variable when a new table is initialized

Parameters:

tbSchema schema of table

tbName table name

Definition at line 214 of file pgtable.php.

References actions, copy(), db, edit(), form, insert(), perms, search(), and view().

Referenced by add_table().

4.4.3.30 Pgtable::set_perms_priority (*\$ tbSchema, \$ tbName*)

set default perms and priority number of a table

Parameters:

schema schema of table

table table name

Definition at line 168 of file pgtable.php.

References db, indexPriority, and sort_field().

Referenced by add_table().

4.4.3.31 Pgtable::set_priority (*\$ priority = 0, \$ schema = false, \$ table = false, \$ field = false*)

set the priority of a column and reorder columns

Parameters:

priority integer vvalue

schema schema of table

table table name

field column name

Definition at line 152 of file pgtable.php.

References indexPriority, priority, and table.

Referenced by sort_field().

4.4.3.32 Pgtable::sort_field ()

reorder the table, how \$indexPriority specify

Definition at line 139 of file pgtable.php.

References set_priority().

Referenced by set_perms_priority().

4.4.3.33 Pgtable::submit (\$ tbSchema, \$ tbName, \$ arr)

Definition at line 1728 of file pgtable.php.

4.4.3.34 Pgtable::switch_actions (\$ tbSchema = 'public', \$ tbName)

main function

Parameters:

tbSchema= 'public' schema name

tbName table name

Returns:

a hash with keys: text: text to display at the bottom of the table search_bar: search bar submit: submit buttons table: html table

Definition at line 1519 of file pgtable.php.

References arr_to_hidden(), html_table(), and search_bar().

4.4.3.35 Pgtable::switch_edit (\$ tbSchema, \$ tbName)

Definition at line 1676 of file pgtable.php.

4.4.3.36 Pgtable::update (\$ tbSchema, \$ tbName)

called when the html button 'Save' is clicked (only save modified data)

Parameters:

tbSchema schema name of the table

tbName table name

Definition at line 487 of file pgtable.php.

4.4.3.37 Pgtable::view (\$ tbSchema, \$ tbName)

called when the html button 'View' is clicked

Parameters:

tbSchema schema name of the table

tbName table name

Returns:

bidimensional array, that corespond to html table

Definition at line 612 of file phtable.php.

Referenced by auto_join_query(), join_parent(), and set_defaults().

4.4.3.38 Pgtable::view_edit (\$ tbSchema, \$ tbName, \$ when, &\$ datain)

called from others functions

Parameters:

tbSchema schema name of the table

tbName table name

when when is called (new, edit, search,...)

datain input data (\$_POST or \$_GET)

Returns:

bidimensional array, that corespodnd to html table

Definition at line 669 of file phtable.php.

References arr_to_hidden(), and table.

4.4.3.39 Pgtable::view_sub (\$ tbSchema, \$ tbName, \$ field, \$ num, \$ when, \$ def)

called for each field, to view its value in an appropriate parsed mode

Parameters:

tbSchema schema name of the table

tbName table name

field column name

num line number of current sets of records

when when is calles(search, edit, insert...)

def current value of the field

Definition at line 1311 of file phtable.php.

4.4.4 Member Data Documentation**4.4.4.1 Pgtable::actions**

Definition at line 58 of file phtable.php.

Referenced by set_defaults().

4.4.4.2 Pgtable::configSchema

Definition at line 62 of file phtable.php.

Referenced by Pgtable().

4.4.4.3 Pgtable::data

Definition at line 55 of file pgtable.php.

Referenced by Pgtable().

4.4.4.4 Pgtable::db

Definition at line 46 of file pgtable.php.

Referenced by add_table(), auto_join(), auto_join_query(), Pgtable(), set_defaults(), and set_perms_priority().

4.4.4.5 Pgtable::errors

Definition at line 68 of file pgtable.php.

Referenced by Pgtable().

4.4.4.6 Pgtable::form

Definition at line 54 of file pgtable.php.

Referenced by auto_join(), auto_join_query(), Pgtable(), and set_defaults().

4.4.4.7 Pgtable::group

Definition at line 61 of file pgtable.php.

Referenced by Pgtable().

4.4.4.8 Pgtable::indexPriority

Definition at line 66 of file pgtable.php.

Referenced by set_perms_priority(), and set_priority().

4.4.4.9 Pgtable::inh

Definition at line 50 of file pgtable.php.

Referenced by add_table().

4.4.4.10 Pgtable::links

Definition at line 56 of file pgtable.php.

4.4.4.11 Pgtable::messages

Definition at line 69 of file pgtable.php.

Referenced by Pgtable().

4.4.4.12 Pgtable::operator

Definition at line 64 of file pgtable.php.

4.4.4.13 Pgtable::perms

Definition at line 51 of file pgtable.php.

Referenced by Pgtable(), and set_defaults().

4.4.4.14 Pgtable::pkey

Definition at line 48 of file pgtable.php.

Referenced by add_table(), auto_join(), and auto_join_query().

4.4.4.15 Pgtable::priority

Definition at line 52 of file pgtable.php.

Referenced by Pgtable(), and set_priority().

4.4.4.16 Pgtable::ref

Definition at line 49 of file pgtable.php.

Referenced by add_table(), alter_parent(), auto_join(), auto_join_query(), join_child(), and join_parent().

4.4.4.17 Pgtable::STRINGS

Definition at line 70 of file pgtable.php.

Referenced by auto_join(), and Pgtable().

4.4.4.18 Pgtable::user

Definition at line 60 of file pgtable.php.

Referenced by Pgtable().

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

- [pgtable.php](#)

Chapter 5

phpPgWeb File Documentation

5.1 config.ini File Reference

Variables

- libraries directory `INCLUDE_DIR = /home/paolo/www/phpPgWeb2/include/libs/`

5.1.1 Variable Documentation

5.1.1.1 libraries directory `INCLUDE_DIR = /home/paolo/www/phpPgWeb2/include/libs/`

Definition at line 3 of file config.ini.

5.2 config.php File Reference

5.3 general.php File Reference

Functions

- [array_to_hash](#) (\$arr1,\$arr2)
convert 2 array as an hash having values of 1th array as key for values of second array
- [transpose](#) (\$arr)
transpose a matrix
- [isodate](#) (\$datearr)
isodate
- [is_date](#) (\$datearr)
is_date
- [isotime](#) (\$timearr)
isotime
- [df](#) (\$date="", \$fmt=false)
return a date string
- [o_iftrue](#) (\$fmt,\$str)
return a string formatted as defined if \$str is not null
- [clean_exif](#) (&\$arr)
clean an array from binary values
- [exif_to_array](#) (\$exif)
manage exif data from jpeg/tiff image, from the db
- [array_to_exif](#) (\$exif_array)
manage exif data from jpeg/tiff image, to insert into a db
- [secure_unset](#) (\$name)
- [array_unset](#) (&\$array,\$keys)
- [getmicrotime](#) ()

5.3.1 Function Documentation

5.3.1.1 [array_to_exif](#) (\$ exif_array)

manage exif data from jpeg/tiff image, to insert into a db

NOT USED!!

Parameters:

exif

Definition at line 212 of file general.php.

5.3.1.2 array_to_hash (\$ arr1, \$ arr2)

convert 2 array as an hash having values of 1th array as key for values of second array

Parameters:

arr1 array to use as keys

arr2 array of values

Returns:

associative array

Definition at line 29 of file general.php.

5.3.1.3 array_unset (&\$ array, \$ keys)

Definition at line 234 of file general.php.

Referenced by Pgtable::delete(), and Pgtable::search().

5.3.1.4 clean_exif (&\$ arr)

clean an array from binary values

it is used to clean exif results from jpeg/tiff images. It clean \$arr from bad insignificant binary values to bestored in a db without errors

Parameters:

& \$arr multidimensional array

Returns:

parsed &\$arr

Definition at line 166 of file general.php.

5.3.1.5 df (\$ date = "", \$ fmt = false)

return a date string

Not used, bad for global variable \$SQL_DATE_FMT return date string (actual if not defined, else in \$SQL_DATE_FMT format(ymd, mdy, dmy)) in it_IT format

Parameters:

date= " date, default current in

fmt=false , *format* of date as strftime function (es : 'x', that is the default)

Returns:

date

Definition at line 123 of file general.php.

5.3.1.6 `exif_to_array` (*\$ exif*)

manage exif data from jpeg/tiff image, from the db
NOT USED!!

Parameters:

exif

Definition at line 188 of file general.php.

5.3.1.7 `getmicrotime` ()

Definition at line 247 of file general.php.
Referenced by Pgtable::search().

5.3.1.8 `is_date` (*\$ datearr*)

is_date

Parameters:

datearr

Returns:

checkdate

Definition at line 90 of file general.php.

5.3.1.9 `isodate` (*\$ datearr*)

isodate

Parameters:

datearr

Returns:

isodate

Definition at line 75 of file general.php.

References isodate().

Referenced by isodate().

5.3.1.10 `isotime` (*\$ timearr*)

isotime

Parameters:

timearr

Returns:

isotime

Definition at line 101 of file general.php.

References isotime().

Referenced by isotime().

5.3.1.11 `o_iftrue ($fmt, $str)`

return a string formatted as defined if \$str is not null

Parameters:

fm format

str string

Returns:

string

Definition at line 147 of file general.php.

5.3.1.12 `secure_unset ($name)`

Definition at line 225 of file general.php.

5.3.1.13 `transpose ($arr)`

transpose a matrix

Parameters:

arr bidimensional array

Returns:

transpose array

Definition at line 50 of file general.php.

Referenced by `html_table()`, and `Pgtable::search()`.

5.4 html.php File Reference

Compounds

- class [Menu](#)
class to manage menu

Functions

- [js_sanitize](#) (\$string)
clean backslashes in javascript sources
- [array_to_html](#) (\$o,\$space=0,\$keys=1,\$encode=true)
convert a multidimensional array into a html list
- [html_select](#) (\$name,\$data,\$default=false,\$assoc=1,\$padding=0,\$mustdefine=1,\$onchange="")
create an html select from an array
- [history_replace](#) (\$url)
clean browser history
- [var_dump_array](#) (\$arr,&\$out,\$var_name="")
convert the input multidimensional array in a bidimensional array that contains in ["name"] php_name of the variable and in ["value"] his value;
- [arr_to_hidden](#) (\$arr,\$baseName="")
return a string of input html tags (for POST query) from an input bidimensional array
- [arr_to_url](#) (\$arr,\$baseName="")
return a string of urlencoded parameters (for GET query) from an input bidimensional array
- [out](#) (\$string,\$stdout=true)
- [html_table](#) (\$arr,\$headerCol=1,\$headerRow=0,\$noInterpolate=0,\$headerColNoInterpolate=0,\$headerRowNoInterpolate=0,\$noInt.type=0,\$transpost=0,\$stdout=true)
print a html table from a bidimensional array
- [search_bar](#) (\$searchUrl,\$previousUrl,\$nextUrl,\$otherUrl,\$total,\$offset=1,\$limit=10,\$numPages=10,\$nameOffset="searchOffset")
create a index bar for search engine
- [date_select](#) (\$fieldname,\$default="", \$mustdefine=1,\$locale="it_IT",\$years="")
date_select
- [time_select](#) (\$fieldname,\$default=false,\$mustdefine=1)
time_select
- [JavaScript](#) (\$code="")
store javascript code in a static variable

- **Alert** (\$message="")
store alert messages in a static variable
- **Error** (\$message="",\$goBack=0)
store error messages in a static array
- **HistoryGo** (\$page="")
store history in a static var
- **WriteJavascript** ()
write javascript code generate from JavaScript(), Alert(), HistoryGo(), Error() functions
- **html_head** (\$code="")
store html head code in a static variable and return all
- **postgis_forinput** (\$fieldname,\$default="",\$type='POINT')
crate an html input

5.4.1 Function Documentation

5.4.1.1 Alert (\$ message = "")

store alert messages in a static variable

Parameters:

message

Returns:

messages

Definition at line 519 of file html.php.

Referenced by WriteJavascript().

5.4.1.2 arr_to_hidden (\$ arr, \$ baseName = "")

return a string of input html tags (for POST query) from an input bidimensional array

Return a string of input html tags (for POST query) from an input bidimensional array in wich the second dimension ["name"] indicate the name of the variable and the second dimension["value"] indicate the value of the variable

Parameters:

arr bidimensional array in wich the second dimension ["name"] indicate the name of the variable and the second dimension["value"] indicate the value of the variable

basename= " base name of all input names

Returns:

\$src html code

Definition at line 152 of file html.php.

References out(), and var_dump_array().

Referenced by search_bar(), Pgtable::switch_actions(), and Pgtable::view_edit().

5.4.1.3 arr_to_url (\$arr, \$baseName = '')

return a string of urlencoded parameters (for GET query) from an input bidimensional array

Return a string of urlencoded parameters (for GET query) from an input bidimensional array in wich the second dimension ["name"] indicate the name of the variable and the second dimension["value"] indicate the value of the variable

Parameters:

arr bidimensional array in wich the second dimension ["name"] indicate the name of the variable and the second dimension["value"] indicate the value of the variable

basename= '' base name of all input names

Returns:

\$url urlencoded

Definition at line 174 of file html.php.

References out(), and var_dump_array().

Referenced by Pgtable::order().

5.4.1.4 array_to_html (\$o, \$space = 0, \$keys = 1, \$encode = true)

convert a multidimensional array into a html list

Parameters:

o array

space number of space to indent

keys if display keys array: 1 if all, 2 if all except the last dimension of array, 0 for not

encode=true if use htmlentities

Returns:

html code

Definition at line 47 of file html.php.

5.4.1.5 date_select (\$fieldname, \$default = '', \$mustdefine = 1, \$locale = 'it_IT', \$years = '')

date_select

Parameters:

fieldname

default

mustdefine=1

locale=it_IT

years= " array of years, default 5 years befor and after current

Returns:

html code

Definition at line 438 of file html.php.

References html_select().

5.4.1.6 Error (\$ message = "", \$ goBack = 0)

store error messages in a static array

Parameters:

message= " errore

goBack=0 if call HistoryGo(-1);

Returns:

errors array

Definition at line 533 of file html.php.

References HistoryGo().

Referenced by WriteJavascript().

5.4.1.7 history_replace (\$ url)

clean browser history

Parameters:

url url to go after cleaning

Returns:

html code

Author:

Antonio Galea

Definition at line 104 of file html.php.

5.4.1.8 HistoryGo (\$ page = "")

store history in a static var

Parameters:

page= " page

Returns:

var

Definition at line 546 of file html.php.

References HistoryGo().

Referenced by Error(), HistoryGo(), and WriteJavascript().

5.4.1.9 `html_head` ($\$code = ''$)

store html head code in a static variable and return all

Parameters:

code

Returns:

head code

Definition at line 580 of file html.php.

5.4.1.10 `html_select` ($\$name, \$data, \$default = \text{false}, \$assoc = 1, \$padding = 0, \$mustdefine = 1, \$onchange = ''$)

create an html select from an array

Parameters:

name name of select form

data array of data

default= " selected value

assoc=1 if it is an associative array (name, value)

padding=0 indentation

mustdefined=1 if not add a void element

onchange= " javascript code

Returns:

html code

Definition at line 76 of file html.php.

References `js_sanitize()`.

Referenced by `date_select()`, `Pgtable::operator()`, and `time_select()`.

5.4.1.11 `html_table` ($\$arr, \$headerCol = 1, \$headerRow = 0, \$noInterpolate = 0, \$headerColNoInterpolate = 0, \$headerRowNoInterpolate = 0, \$noInt_type = 0, \$transpost = 0, \$stdout = \text{true}$)

print a html table from a bidimensional array

NOTE: not write table tags

Parameters:

arr associative bidimensional array

headerCol=1 if make header line using keys of \$arr, 2 if use first element

headerRow=1 if make header column using keys of \$arr, 2 if use first element

noInterpolate=0 if not interpolate \$arr values with `htmlentities` function. Note that this could be 1 or 0 and so it refers itself to all the input array, or a monodimensional array with a values for any

arr:

column if

noInt_type=0, line if \$noInt_type=1 or a bidimensional array with a values for any \$arr elemnt. If not all values are defined, default is interpolated.

Parameters:

headerColNoInterpolate as \$noInterpolate but for header

headerRowNoInterpolate as \$noInterpolate but for header

noInt_type=0, see up

transpost=0 if matrix \$arr must be transponded

stdout=true if print table directly to stdout or return html code

Definition at line 210 of file html.php.

References out(), and transpose().

Referenced by Pgtable::switch_actions().

5.4.1.12 JavaScript (\$ code = "")

store javascript code in a static variable

Parameters:

code

Returns:

javascript code

Definition at line 507 of file html.php.

Referenced by WriteJavascript().

5.4.1.13 js_sanitize (\$ string)

clean backslashes in javascript sources

Parameters:

string javascript code

Returns:

javascript code

Author:

Antonio Galea

Definition at line 30 of file html.php.

Referenced by html_select().

5.4.1.14 out (\$ string, \$ stdout = true)

Definition at line 184 of file html.php.

Referenced by arr_to_hidden(), arr_to_url(), html_table(), and var_dump_array().

5.4.1.15 postgis_forminput (*\$fieldname*, *\$default = ''*, *\$type = 'POINT'*)

crate an html input

Definition at line 591 of file html.php.

5.4.1.16 search_bar (*\$searchUrl*, *\$previousUrl*, *\$nextUrl*, *\$otherUrl*, *\$total*, *\$offset = 1*, *\$limit = 10*, *\$numPages = 10*, *\$nameOffset = "searchOffset"*)

create a index bar for search engine

Parameters:

searchUrl the url to link(whit oll the parameters urlencoded requiered by the search script). Note that this function add automaticly searchOffset paramater; this if you need a get.... if you would use post method \$searchUrl must be an array (key=name of hidden input, value=value...).

nextUrl url of image for next page

previousUrl url of image for previous page

otherUrl url of image for other pages

total the total records number

offset the record number from which it starts to display this page (it is at minimum 1). If not set, default is 1

limit the number of results to display, default is 10

numPages the number of pages to display into the index bar Default is 10

Returns:

html code

Definition at line 365 of file html.php.

References arr_to_hidden(), and title.

Referenced by Pgtable::switch_actions().

5.4.1.17 time_select (*\$fieldname*, *\$default = false*, *\$mustdefine = 1*)

time_select

Parameters:

fieldname

default

mustdefine=1

Returns:

html code

Definition at line 479 of file html.php.

References html_select().

5.4.1.18 `var_dump_array` (`$ arr`, `&$ out`, `$ var_name = ''`)

convert the input multidimensional array in a bidimensional array that contains in [”name”] php_name of the variable and in [”value”] his value;

Note: it works also if \$arr is not an array

Parameters:

arr input array

& \$out the output array

var_name the base name of the variable, default ”

Definition at line 128 of file html.php.

References out().

Referenced by arr_to_hidden(), and arr_to_url().

5.4.1.19 `WriteJavascript` ()

write javascript code generate from [JavaScript\(\)](#), [Alert\(\)](#), [HistoryGo\(\)](#), [Error\(\)](#) functions

Definition at line 557 of file html.php.

References [Alert\(\)](#), [Error\(\)](#), [HistoryGo\(\)](#), and [JavaScript\(\)](#).

5.5 index.php File Reference

Variables

- [PgDbConn](#)
- `table_list = get_tables($PgDbConn)`
- `tables = array_to_html($tables,0,2,false)`
- `title = "table lists"`

5.5.1 Variable Documentation

5.5.1.1 PgDbConn

Initial value:

```
pg_connect(
    "dbname=" . $CONFIG["DB"][DBNAME] .
    " user=" . $CONFIG["DB"]['USER'] .
    " password=" . $CONFIG["DB"][PASSWORD] .
    " host=" . $CONFIG["DB"]['HOST']
)
or die("Can't connect!")
```

Definition at line 6 of file index.php.

5.5.1.2 `table_list = get_tables($PgDbConn)`

Definition at line 16 of file index.php.

5.5.1.3 `tables = array_to_html($tables,0,2,false)`

Definition at line 24 of file index.php.

5.5.1.4 `title = "table lists"`

Definition at line 25 of file index.php.

Referenced by `Pgtable::actions()`, `Pgtable::order()`, and `search_bar()`.

5.6 init.php File Reference

Variables

- `CONFIG = parse_ini_file("../conf/config.ini",true)`

5.6.1 Variable Documentation

5.6.1.1 `CONFIG = parse_ini_file("../conf/config.ini",true)`

Definition at line 25 of file init.php.

5.7 lobjview.php File Reference

Variables

- [PgDbConn](#)
- `lo = pg_lo_open($PgDbConn,$_GET['loid'],'r')`

5.7.1 Variable Documentation

5.7.1.1 `lo = pg_lo_open($PgDbConn,$_GET['loid'],'r')`

Definition at line 15 of file lobjview.php.

5.7.1.2 `PgDbConn`

Initial value:

```
pg_connect(
    "dbname=" . $CONFIG[ "DB" ][ DBNAME ] .
    " user=" . $CONFIG[ "DB" ][ 'USER' ] .
    " password=" . $CONFIG[ "DB" ][ PASSWORD ] .
    " host=" . $CONFIG[ "DB" ][ 'HOST' ]
)
or die("Can't connect!")
```

Definition at line 4 of file lobjview.php.

5.8 main.dox File Reference

5.9 pgdb.php File Reference

Compounds

- class [Pgdb](#)
this class is used to interface [Pgtable](#) class to postgresql,

5.10 pgmessage.php File Reference

Compounds

- class [Pgmessages](#)

5.11 pgtable.php File Reference

Compounds

- class [Pgtable](#)
the main class that is used to create the web interface to a postgresql database

5.12 pgtbconf.ini File Reference

Variables

- ini file that contain translation of used strings change to change language `strings = /home/paolo/www/phpPgWeb2/conf/strings.en`
- where files are stored [DbFiles]
- remote visible directory trough web server `remote = /dbFiles/`
- local directory in wich php store uploaded files `local`

5.12.1 Variable Documentation

5.12.1.1 local directory in wich php store uploaded files local

Initial value:

```
/home/paolo/www/phpPgWeb2/html/dbFiles/  
  
[MimeTypeIcons]  
  
remote=/image/mime-type/  
local=/home/paolo/www/phpPgWeb2/html/image/mime-type/  
  
[html]
```

Definition at line 11 of file pgtbconf.ini.

5.12.1.2 remote visible directory trough web server remote = /dbFiles/

Definition at line 8 of file pgtbconf.ini.

5.12.1.3 where files are stored[DbFiles]

Definition at line 7 of file pgtbconf.ini.

5.12.1.4 ini file that contain translation of used strings change to change language strings = /home/paolo/www/phpPgWeb2/conf/strings.en

Definition at line 2 of file pgtbconf.ini.

5.13 postgres.php File Reference

Functions

- **`dati_sql`** (`$array`, `$not_null=0`, `$is_expression=0`, `$dati=""`, `$default_not_null=0`)
format sql variables
- **`insert_sql`** (`$tbSchema='public'`, `$table_name`, `$array`, `$not_null=0`, `$is_expression=0`, `$dati=""`, `$default_not_null=0`)
format sql insert query
- **`update_sql`** (`$tbSchema='public'`, `$table_name`, `$array`, `$not_null=0`, `$is_expression=0`, `$dati=""`, `$default_not_null=0`)
format sql update query
- **`where_sql`** (`$array`, `$operator='='`, `$logic='AND'`, `$tbSchema='public'`, `$tbName=""`, `$not_null=0`, `$is_expression=0`, `$dati=""`, `$default_not_null=0`)
format sql where query
- **`isopostgis`** (`$coord`, `$type`)
create postig type from array of coordinates and geometry type
- **`get_table_attributes`** (`$conn`, `$table`, `$field=""`, `$result_type1=PGSQL_ASSOC`, `$result_type2=PGSQL_BOTH`, `$namespace='public'`)
Retrieve the attribute definition of a table.

5.13.1 Function Documentation

5.13.1.1 `dati_sql` (`$ array`, `$ not_null = 0`, `$ is_expression = 0`, `$ dati = ""`, `$ default_not_null = 0`)

format sql variables

formatta i dati aggiungendo apici se servono o settando a NULL le variabili vuote, o lasciandole vuote se col29i specificato dal terzo parametro della funzione

Parameters:

`array` sql data

`not_null=0` var or array that indicates if all or a variable could be a void string or must be NULL

`dati=""` name of key of \$array to parse (default all)

`default_not_null=0` if you set \$could_be_void as an array, but not for all the values, this set remanent values

`is_expression=0` var or array that indicates if all or a variable is a sql expression or a value to be enclose into 2 ""

Returns:

parsed array

Definition at line 35 of file postgres.php.

Referenced by `insert_sql()`, `update_sql()`, and `where_sql()`.

5.13.1.2 `get_table_attributes` (*\$conn*, *\$table*, *\$field = ''*, *\$result_type1 = PGSQL_ASSOC*, *\$result_type2 = PGSQL_BOTH*, *\$namespace = 'public'*)

Retrieve the attribute definition of a table.

Parameters:

conn connection resource

table The name of the table

field (optional) The name of a field to return

result_type1 one of predef. constant PGSQL_BOTH, PGSQL_ASSOC , PGSQL_NUM for `pg_query()` function (for first dimension key of array)

result_type2 one of predef. constant PGSQL_BOTH, PGSQL_ASSOC , PGSQL_NUM for `pg_query()` function (for second dimension key of array)

namespace= 'public' name of postgres schema

Returns:

associative array attributes in order

Definition at line 223 of file `postgres.php`.

References table.

5.13.1.3 `insert_sql` (*\$tbSchema = 'public'*, *\$table_name*, *\$array*, *\$not_null = 0*, *\$is_expression = 0*, *\$dati = ''*, *\$default_not_null = 0*)

format sql insert query

formatta i dati aggiungendo apici se servono o settando a NULL le variabili vuote, o lasciandole vuote se co129i specificato dal terzo parametro della funzione

Parameters:

array sql data

table_name

not_null=0 var or array that indicates if all or a variable could be a void string or must be NULL

dati= '' name of key of \$array to insert (default all)

default_not_null=0 if you set \$not_null as an array, but not for all the values, this set remanent values

is_expression=0 var or array that indicates if all or a variable is a sql expression or a value to be enclose into 2 ''

Returns:

parsed sql query

Definition at line 92 of file `postgres.php`.

References `dati_sql()`.

Referenced by `Pgdb::insert()`.

5.13.1.4 isopostgis (\$ coord, \$ type)

create postgis type from array of coordinates and geometry type

Parameters:

coord bidimensional array [east] and [north] or monodimensional if type is point

Returns:

postgis string

Definition at line 198 of file postgres.php.

5.13.1.5 update_sql (\$ tbSchema = 'public', \$ table_name, \$ array, \$ not_null = 0, \$ is_expression = 0, \$ dati = "", \$ default_not_null = 0)

format sql update query

formatta i dati aggiungendo apici se servono o settando a NULL le variabili vuote, o lasciandole vuote se col129i specificato dal terzo parametro della funzione

Parameters:

array sql data

table_name

not_null=0 var or array that indicates if all or a variable could be a void string or must be NULL

dati= " name of key of \$array to parse (default all)

default_not_null=0 if you set \$not_null as an array, but not for all the values, this set remanent values

is_expression=0 var or array that indicates if all or a variable is a sql expression or a value to be enclose into 2 "

Returns:

parsed sql query

Definition at line 128 of file postgres.php.

References dati_sql().

Referenced by Pgdb::update().

5.13.1.6 where_sql (\$ array, \$ operator = '=', \$ logic = 'AND', \$ tbSchema = 'public', \$ tbName = "", \$ not_null = 0, \$ is_expression = 0, \$ dati = "", \$ default_not_null = 0)

format sql where query

formatta i dati aggiungendo apici se servono o settando a NULL le variabili vuote, o lasciandole vuote se col129i specificato dal terzo parametro della funzione

Parameters:

array sql data array

operator= '=' one operator for all or associative array of operators

logic= 'AND' logic opertor to use

not_null=0 var or array that indicates if all or a variable could be a void string or must be NULL

dati= " name of key of \$array to parse (default all)

tbName= " var or array of table names of all or each field(used in query with more fields of different columns that have the same name)

default_not_null=0 if you set \$not_null as an array, but not for all the values, this set remanent values

is_expression=0 var or array that indicates if all or a variable is a sql expression or a value to be enclose into 2 "

Returns:

parsed sql query

Definition at line 162 of file postgres.php.

References `dati_sql()`.

Referenced by `Pgdb::delete()`, `Pgtable::search()`, and `Pgdb::update()`.

5.14 retrocompatibilita.php File Reference

Functions

- [pg_num_rows](#) (\$result)
- [pg_num_fields](#) (\$result)
- [pg_free_result](#) (\$result)
- [pg_lo_import](#) (\$connection,\$result)
- [pg_lo_read_all](#) (\$result)
- [pg_lo_read](#) (\$result,\$size)
- [pg_lo_close](#) (\$result)
- [pg_lo_open](#) (\$conn,\$result,\$mode)
- [pg_last_error](#) ()
- [pg_affected_rows](#) (\$q)
- [exif_read_data](#) (\$image,\$bo1=1,\$bo2=2,\$thumbnail=0)
- [pg_escape_string](#) (\$result)

Variables

- return [array_out](#)

5.14.1 Function Documentation

5.14.1.1 [exif_read_data](#) (\$ *image*, \$ *bo1* = 1, \$ *bo2* = 2, \$ *thumbnail* = 0)

Definition at line 70 of file retrocompatibilita.php.

Referenced by Pgdb::readExif().

5.14.1.2 [pg_affected_rows](#) (\$ *q*)

Definition at line 63 of file retrocompatibilita.php.

Referenced by Pgdb::delete(), Pgdb::insert(), and Pgdb::update().

5.14.1.3 [pg_escape_string](#) (\$ *result*)

Definition at line 81 of file retrocompatibilita.php.

Referenced by Pgtable::auto_join_query().

5.14.1.4 [pg_free_result](#) (\$ *result*)

Definition at line 36 of file retrocompatibilita.php.

5.14.1.5 [pg_last_error](#) ()

Definition at line 60 of file retrocompatibilita.php.

5.14.1.6 pg_lo_close (\$ result)

Definition at line 52 of file retrocompatibilita.php.

5.14.1.7 pg_lo_import (\$ connection, \$ result)

Definition at line 40 of file retrocompatibilita.php.

5.14.1.8 pg_lo_open (\$ conn, \$ result, \$ mode)

Definition at line 56 of file retrocompatibilita.php.

5.14.1.9 pg_lo_read (\$ result, \$ size)

Definition at line 48 of file retrocompatibilita.php.

5.14.1.10 pg_lo_read_all (\$ result)

Definition at line 44 of file retrocompatibilita.php.

5.14.1.11 pg_num_fields (\$ result)

Definition at line 32 of file retrocompatibilita.php.

5.14.1.12 pg_num_rows (\$ result)

Definition at line 28 of file retrocompatibilita.php.

5.14.2 Variable Documentation**5.14.2.1 return array_out**

Definition at line 93 of file retrocompatibilita.php.

5.15 tables.php File Reference

Variables

- [PgDbConn](#)
- `table = new Pgtable($PgDbConn, "../conf/pgtbconf.ini")`

5.15.1 Variable Documentation

5.15.1.1 PgDbConn

Initial value:

```
pg_connect(
    "dbname=" . $CONFIG["DB"][DBNAME] .
    " user=" . $CONFIG["DB"]['USER'] .
    " password=" . $CONFIG["DB"][PASSWORD] .
    " host=" . $CONFIG["DB"]['HOST']
)
or die("Can't connect!")
```

Definition at line 6 of file tables.php.

5.15.1.2 `table = new Pgtable($PgDbConn, "../conf/pgtbconf.ini")`

Definition at line 16 of file tables.php.

Referenced by [Pgtable::alter_display\(\)](#), [Pgtable::auto_join\(\)](#), [Pgtable::auto_join_query\(\)](#), [Pgdb::check-UploadedFile\(\)](#), [Pgdb::get_foreign_keys\(\)](#), [get_table_attributes\(\)](#), [Pgtable::join_child\(\)](#), [Pgtable::join-parent\(\)](#), [Pgtable::parse_data_in\(\)](#), [Pgtable::set_priority\(\)](#), [Pgdb::set_value_from_pkey\(\)](#), and [Pgtable::view-edit\(\)](#).