

pgDesigner

Database Designer for PostgreSQL

User Manual

rev. 26/11/2007

rif. vers. 1.1.0

Index

Preface.....	4
Introduction.....	6
The Application.....	8
The Main Window.....	9
Menus.....	12
The Main Menu.....	12
Popup Menus.....	18
The Menu Project.....	18
The Menu Edit.....	18
The Menu Bar.....	19
The Panel Tools.....	21
The Status Panel.....	22
The Panel Projects.....	24
Area of Design / Diagram.....	24
The View List.....	24
The Object List.....	25
Project.....	26
Create a New Project.....	27
Create a Project importing structure from a database (reverse-engineering).....	28
Opening an existing project.....	30
File Loading.....	30
Uploading files from Sql.....	30
Export Project.....	31
Exports to file Sql.....	32
Exports on Database.....	32
Saving Project.....	34
Printing the Diagram.....	35
Preview.....	35
Print.....	36
Image File Creation.....	36
Reporting.....	37
Documentation in HTML format.....	37
Documentation in PDF format.....	38
Documentation in TXT format.....	39
Project View / Display.....	40
Main View.....	40
Creating Custom Views.....	40
Managing objects in a customized view.....	41
Entities and Objects.....	43
Creation.....	44
Dimensions.....	44
Movement.....	44

Editing.....	44
Deleting.....	44
Copy on another project.....	44
Description of Objects.....	46
Area.....	47
Domain.....	48
Procedure/Function.....	50
Relation.....	53
Sequence.....	55
Stamp.....	56
Table.....	57
Tablespace.....	63
Text.....	65
Trigger.....	66
View.....	68
View Creation Wizard.....	71
Other functions.....	75
Configuring pgDesigner.....	76
The window Statistics.....	81
Changing location coordinates of an object.....	85
Customizing color of an object.....	86
Finding objects within a project.....	87
Viewing of the log.....	88
Serious errors.....	89
General Configuration.....	90
The configuration file.....	90
Structure of configuration files.....	91
The File Project.....	94
Structure File Project.....	95
INI Format.....	95
XML Format.....	104
License.....	112
GNU GENERAL PUBLIC LICENSE.....	113

Preface

pgDesigner is an open source program for graphic design database to *PostgreSQL*. The code is written in the language *Gambas*, and currently runs only on Linux operating system.

Currently, the *Gambas* language comes in two versions: stable and development. Because of its advanced features than the stable for the construction of *pgDesigner* was used the second.

pgDesigner is in a state of continuous development and updating, given the continuing evolution of *Gambas* and *PostgreSQL*, both continuous technological implementations required by the continuing evolution of software in general.

Although it is constantly evolving, *pgDesigner* can be safely used as a basis for building database engine based on *PostgreSQL*.

At present are implemented the following features:

- Management multiproject.
- Management views of the project.
- Copying objects between different projects.
- Creating object-based *PostgreSQL*, such as tables, fields and indexes, views, relations, tablespaces, procedures, triggers, types, domains and sequences.
- Creating areas delimitation rectangular, with customization of color fill and text.
- Creating text as graphics in the diagram of the project.
- Saving files on projects, both in format INI and XML.
- Managing release of *PostgreSQL*, from version 7.0 to 8.2.
- Reverse-engineering conducted by *PostgreSQL* servers, local or remote.
- Customizing visual environment in general, the basic settings of individual project or individual object, management of visual details.
- Management of the project on a diagram chart.
- Automatic updating of relations between objects of the project.
- Each object type distinct from special icon.
- Placement of objects by dragging with the mouse or through manual insertion coordinates.
- Automatic updating of relations between tables.
- Wizard for the construction of views.
- Panel projects, viewing and management of multiple simultaneous projects.
- List of items in each project.
- List of views for each individual project.
- Panel instruments containing objects or entities that can be inserted in the draft.
- State Panel to view the current status of the current project.
- Viewing and write files on SQL commands necessary to create the database.
- Creating direct database on a server *PostgreSQL*.
- Management preview for the layout of the print graphic design, with the possibility of sending printer or image files.

- Management reports project with preview, text files or sending to the printer, in formats: text, html and pdf.
- Customizing global color, or of individual objects.
- Creating projects on uploading files containing SQL commands.
- Function search objects in the project.

To perform *pgDesigner* you must install the entire development environment or sun libraries *Gambas2* (development version), available from the official site:

<http://gambas.sourceforge.net>

The latest version of the program *pgDesigner* can be downloaded at:

<http://sourceforge.net/projects/pgdesigner>.

For users of the Italian language is also available on the Forum website:

<http://www.ldrweb.net>.

Note: Developing *pgDesigner* follows constantly updates *Gambas2*. Often they change some internal features, or implemented new features compared to previous versions of *Gambas2*, so it is necessary to update the libraries of language. In any case, in ChangeLog file to the package *pgDesigner*, will be reported each time the version of *Gambas2* which was compiled *pgDesigner* at the time of issue. Normally, the program comes in compressed format, containing the sources, both in RPM format for the tracks, as the latter has to bear in mind that the content is the result of the compilation of sources with the version of *Gambas2* of the moment, for it is possible that these are incompatible with the operating system, making it necessary to rebuild from the source code.

PostgreSQL is a registered trademark of PostgreSQL Global Development Group-University of California, and released under BSD license.

Gambas is released under the GNU General Public License.

PgDesigner is released under the GNU General Public License.

Introduction

The project *pgDesigner* was designed to provide a graphical environment, open-source and non-commercial, as an aid for the design and creation of database *PostgreSQL*. In effect on the market similar applications exist, and certainly more advanced *pgDesigner*, but are often commercial, non-free or subject to restrictive licensing.

pgDesigner was born after a long and unsuccessful search network of similar programs, and also for my personal bet, because at that I heard about a new language, *Gambas*, which is also open source, and that seems to promise much good as an alternative to Visual Basic in a Linux environment, the thing I was so intrigued by then it will test the real ability and potential, then I asked myself, why not combine the two?

As evidenced by its author (Benoit Misinini) on the official website of [Gambas](#), this language does not want to be absolutely a copy of the Windows environment, and from what I have seen is actually the case. *Gambas*, while maintaining the basic structure of the Basic language, seems to be willing to fix many bugs and shortcomings of its antagonist, while preserving snellezza and power, in addition to the fact that this was born and works under Linux.

After osanna this language, we speak of this manual: *pgDesigner*.

Currently the program has been in a sufficiently functional, with the necessary groundwork for developing projects for the creation of database to *PostgreSQL*. The application is based essentially on a single environment (SDI), with which it is possible to design the structure of an archive.

The development of *PostgreSQL* is constantly changing, especially in recent times where they are well-known RDBMS such as Oracle and DB2, it is possible that some new implementations have not yet been added to the program, but not desperate ...

The interface is very simple and is described in subsequent chapters.

NOTE: This documentation makes reference to the current version of *pgDesigner*, which currently is 1.1.0 and it is possible that some references or images differ later. Normally documentation should follow a step-by-step evolution of the software, but this is possible only in environments medium / large, the number of people is greater than one; unfortunately currently the only developer, tester and manager of the documentation is a only, and this being a non-commercial, developed in the limited time available, in conditions very significant parallel processes.

I thank all those who helped with translations, ideas, advice and various aid.

I invite everyone to contribute to the improvement of the program by providing advice or suggestions.

I am also willing to consider partnerships, both for the environment and development tests, which for the documentation.

Luigi Carlotto

Roma – Italy

e-mail: md9327@mclink.it

The Application

The program *pgDesigner* comes in either source (such as from open-source policy), both in binary format, to allow easy installation, both source and binaries are provided in the form of RPM, whose system is widely used on most Linux distributions.

These packages are available on the site <http://sourceforge.net/projects/pgdesigner/>, updates will be reported through the classic story of the site.

At <http://pgdesigner.sourceforge.net> this is the web of documentation that reflects that provided in this guide, provided in other languages.

For users Italian is also active site <http://www.ldrweb.net>, where there is a forum where you can exchange chat, get information and assistance on *pgDesigner*, or even propose features for improving application.

But now we move to the description of *pgDesigner*.

The execution can be carried out in different ways, depending on type of installation done. As described in the above lines, *pgDesigner* comes in either source binary form (ie ready to run) if the operating system allows the installation of RPM, it becomes quite easy, given that the installation creates is the directory in which the application will reside and related files, is the necessary link in the / bin directory structure of the operating system, and its entry in the menu GUI (eg KDE). If the operating system does not allow this type of installation, then you must manually unpacking the compressed file containing the source of *pgDesigner*, compilation and deployment, these steps are described in the documentation attached to the compressed file.

It should be borne in mind, however, that *pgDesigner* requires mandatory library *Gambas2* you, and you need to run the installation of *Gambas2* before proceeding to install *pgDesigner*.

Once installation is complete, *pgDesigner* can be run from a terminal, run the following command:

```
# ./pgDesigner.gambas
```

With the current version of *pgDesigner* you can start automatically uploading existing projects, in this case it is sufficient to indicate the names of files complete path, following the command, for example:

```
# ./pgDesigner.gambas /home/test.ini /home2/test.xml
```

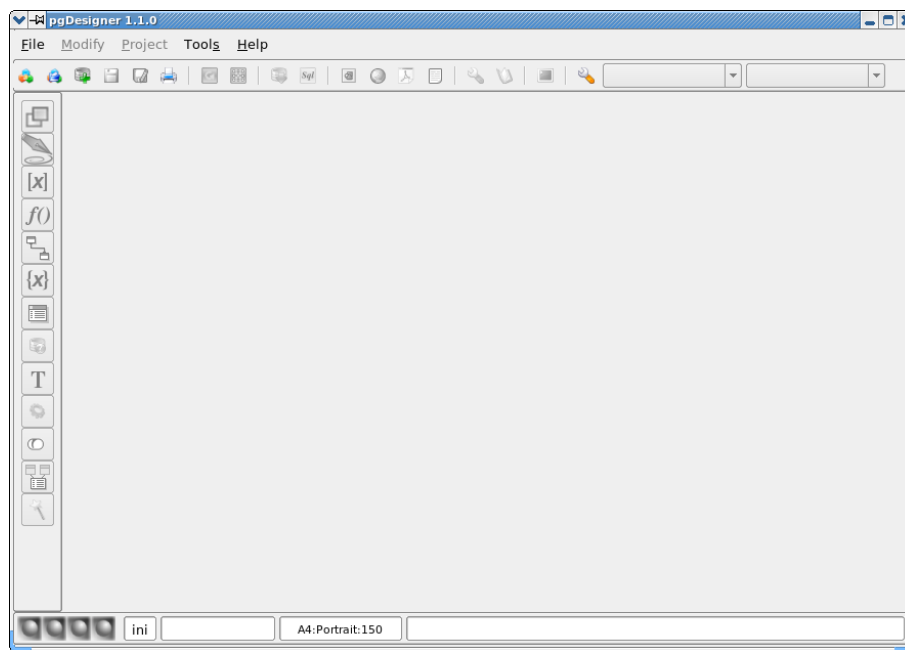
At the end of the boot process, *pgDesigner* launch functions loading files, making them available and visible to the user.

If, as described above, the installation was carried out by the RPM, for the launch of *pgDesigner* just click on the menu item on the desktop.

The Main Window

When, and as you can see from the figure, *pgDesigner* shows a rather simple and rather conforms to the structure of many of the existing applications.

The main menu presents a classic general, and the area below is divided into functional areas. The window can be resized at will, while the various panels can be hidden or resized to allow a broader view of the charts.



In any particular area, or panel, has a specific function, which allows you to control some aspects of a project or view the current status of the project active. On the following page description of these areas.

Description of functional areas

Main Menu Like nearly all existing applications on the market, and not, *pgDesigner* has a main menu containing all the items and options for the management and control of program and project management. As you can see from the preceding page, there are five main elements: File, Edit, Project, Tool and Help; Each of these elements group a number of voices, dividing this type functions and meaning. In subsequent chapters will be described in detail the significance of each item and function within the program.

Menu Bar



To assist the main menu, below this is a horizontal panel, which contains a set of buttons that represent some of the features most useful to the user. On the right are two combo that, when enabled, allows switching between different projects open, or select one of the displays of current project. According to the current status, application or project, the buttons can be disabled, for the items in the main menu.

In chapter 'Menu' is described in detail the function of each button.

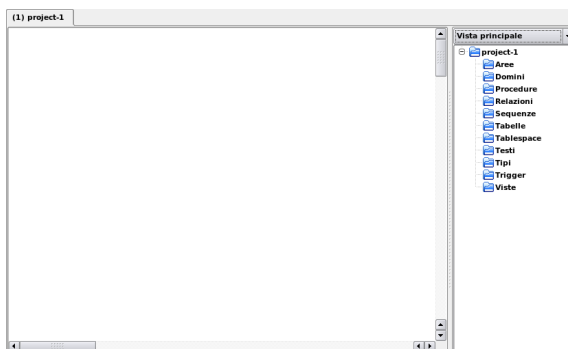
Panel Instruments



As you can see in the illustration on the previous page, to the left of the main window has a vertical bar with a series of buttons with a picture. This panel is called Panel Instruments, and contains buttons necessary to the creation of objects to be included in the project, according to the version of *PostgreSQL* database choice for the project, some of these buttons can be disabled. In subsequent chapters will be described the function of each one of these buttons.

Descrizione delle aree funzionali

Panel projects



In the central area of the main window, there is a panel that displays all projects open each folder in a separate and identifiable by the name of the project in its tongue; projects in the absence of this panel is hidden, showing a completely empty.

Each card contains project is the area on which graphic design scheme of the project, is a list of items contained in the

draft, the list is similarly to the structure of a directory, with all references objects of the project represented by their names and folders contained in sub divided by type. Up to this list is a combo containing views of the project, and with which you can select one for viewing. In subsequent chapters describes the meaning of the concept viewed as a project and manage.

State Panel



At the base of the main window is a visible horizontal panel, which contains graphics and text areas. This panel displays all relevant information on the current status of the project, some of these objects have also feature active. In subsequent chapters will be described features and the type of information displayed each of these areas.

The Menu Bar, the Panel Instruments and the State Panel can be made invisible and in some instances, such as the list of project items, resized. Each variation of the visual is stored in a configuration file General, which will be restored the next time the program; of this file structure will be described in subsequent chapters.





Menus





As mentioned in the preceding paragraph, the application presents a number of menus; compared to the main menu, which contains all the entries and functionality provided by *pgDesigner*, there are other auxiliary menu, including some, such as the menu bar, always visible, while others called Popup Menus are activated and made visible in certain circumstances. These take the form of a drop-down menu with items alone enabled by the current activity is to be noted that the items in these menus are actually blocks of the main menu.

The Main Menu





The menu contains all the necessary items out of the basic functionality of the program and project management. Some of the items listed contain even more items below, in order to regroup so its logical functionality. Some of these elements are connected to specific functions, while others offer more choices by the other are still toggle, or every click reverse their status, making active menu or a particular condition, in this case the state is indicated a check symbol next to the entry.

The entries are grouped by type, and their function is described below:









File	
 New Project	Create an empty project. It gives a dialog for setting the parameters of the project, including the name of the project itself and the version of the database to which you want to see. The confirmation of the data base is created a new card project, including a new list of objects and a new list views.
 Project Open	Get a file previously saved project. It gives a dialog box, so you can select the desired file in the directory of the operating system. The extension of uploaded file extension is set as the default for future rescue. With this feature you can also create a project by reading a file containing SQL statements, in which case it is sufficient to select a file with the extension sql. Files of this type are not considered project files within the application, so the next rescue asked for the name of the file associated with the new project, the original file will no longer be taken into account.
Recent Open	It allows you to select and load a previously saved project files from a list containing the last <n> projects worked, where <n> corresponds to the number set in the general options. The menu appears only when activated options in general and if you have worked at least one file.
 Import from database	Import structure from an existing database, hence creating a new project. It gives a window for the selection of types of objects you want to import, then the window will be displayed for the login and connection to the database server. The loading is done for all of the personal property which has been the connection to the exclusion of anything else, including objects of the system, although present in the same database. According to the version of the <i>PostgreSQL</i> server it is possible that some kind of purpose are not available.
 Save Project	Save the current project files. Currently it is possible to save data in either ini (text),



	Save Project on...	which in XML format, there are no differences between the two formats, but only structural each format also contains the same information as the other for the same project. If the project is new, there is a dialog box to set the new file name or write above a disk already. The extension used for the file name is set as the default for future saves and will no longer be required, unless we use the item described below.
	Close project	Allows saving project on file with a different name and different format. As described in the previous item, now you can save data in either ini (text) format xml. It gives a dialog box to set the filename or write above a disk already. The extension of uploaded files is set as the default for future rescue.
	Print setup...	Closes the current project if the project has undergone changes will be proposed rescuing the latest changes.
	Exit	It allows you to configure the parameters of the press and the default printer to send prints. In any case, you can change these settings later if required (eg printing or graphic reports). The total value settings and customizations override the individual project, with the exception of projects loaded after changes, which maintain the settings saved on the file.








Edit





	Object Copy	Create a copy of the selected object. It gives the dialog box for editing data object. It should be borne in mind that it is not possible to create multiple objects with the same name, so it is likely that is to be amended one proposed for copying. The object created will be added to the current project and displayed located in the upper left of the diagram.
	Delete object	Delete from the project selected object. That operation is sought further confirmation. If you want to delete the object contains references to other objects, they are deleted. The delete function is valid only in the Main View (see charts, views and display project), while if performed in other views of the project, the subject will be deleted only sight, however remaining within the project and displayed in view of Main.
	Edit object	It allows you to edit the properties of the selected object. It gives a dialog to manage its data, leading to the amendment, any changes will be reflected on the record, including cancellation. It should be kept in mind that changes or deletions made by further editing windows activated from the primary dialog box (see edit properties of the fields of a table) will not be recoverable, even if no change is made in the dialog box first.
	Move object	Move the selected object in a different position in the diagram. It gives a dialog box in which you can set manually new X, Y coordinates of the object in the chart, representing the upper left of the diagram and compared to the current display. This feature can also be activated by the State Panel, double-click the mouse in that displays the current position of the object.
	Color object	It allows you to customize the colors of the object selected. It gives a dialog box, for different types of object, then you change the setting of color, including colors that

	can take the object in different states. This customization will be overwritten if the settings are changed in the general options.
Copy on display	It allows you to copy the reference of the selected another view on the same project. Menu Project is a voice with which you can set in a general views and articles related to them, unlike this function is set for the management of a single element, which is the current subject.
Delete from display	It allows to remove the reference of the selected sight active. It should be borne in mind that it is not possible to remove objects from the main view (Main View), for the physical elimination of the project is a voice and a specific function. Menu Project is a voice with which you can set in a general views and articles related to them, unlike this function is set for the management of a single element, which is the current subject. As already described, from the elimination of a different view from that Main does not cause the actual removal from the project, but the mere sight disconnect itself the subject will still be visible in the Main View.
Copy on project	This feature allows you to copy the selected object on another project site. It should be borne in mind that will be copied only the object and its characteristics and properties, with the exception of references to other objects in the project source.
 New	In this menu are a series of related items, with which you can create new objects in the project. The same functionality is exercised, and made easier by Panel tools on the left side of the main window. Enabling or less of items on this menu is dependent on the version of database <i>PostgreSQL</i> chosen for the project.
 Area	It allows you to create an object rectangular. An area can be used to define or highlight a group of other objects, only a purely graphical diagram of the project, not any function or interference with the operations creation of the database. In any condition such items will always drawn without cover most significant objects (eg lines relation ...), then remain as background to the plot. This type of object you can define the fill color and, optionally, a line of text that appears starting from the upper left corner of the internal area, the size and position can be changed at any time.
 Stamp	Create a simple but particular subject. The object is composed of a simple rectangle, which shows some ownership of the project to which it belongs. For a detailed description of this item can be found in later chapters.
 Domain	Create a domain <i>PostgreSQL</i> . It gives a dialog box for entering the necessary data and setting properties. The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official documentation.
 Procedures SQL/PISQL	Create a procedure (or function) <i>PostgreSQL</i> . It gives a dialog box for entering the necessary data and setting properties. The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official documentation.
 Relation	Create a relationship between two tables. The report is actually created after the connection established between the tables, selecting with the mouse, leading to the selection is presented with a dialog box for entering the necessary data and the completion of the properties of relation. To create a relation is that the project must contain at least two tables, complete their respective fields, it is not possible to define relations in the absence of tables and fields. The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official



	Sequence	documentation. Create a sequence <i>PostgreSQL</i> . It gives a dialog box for entering the necessary data and setting properties. The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official documentation.
	Table	Create a table <i>PostgreSQL</i> . It gives a dialog box for entering the necessary data and setting properties; through this window you can also access the editing fields and indexes of the table. It should be borne in mind that some of the data may be entered unless there are related items in the project, for example, you can define the tablespace the project, in this case the combo will contain a reference to this tablespace allowing the hook to the table. The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official documentation.
	Tablespace	Create a tablespace <i>PostgreSQL</i> . It gives a dialog box for entering the necessary data and setting properties. The creation of a tablespace is needed if you want to combine, for example, object to a table. The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official documentation.
T	Text	Create an object text. It gives a dialog box for entering the necessary data and setting properties. As for the areas, the text object has no real function on the database, and is considered only a graphic within the diagram, which may be used as a note or to define areas logical. The string of text entered will be displayed in the diagram to the desired position and can be dragged, modified and deleted as any other object of the project.
	Type	Create a type <i>PostgreSQL</i> . It gives a dialog box for entering the necessary data and setting properties. The types in the project will be presented whenever it is necessary, appear on lists with the types of default <i>PostgreSQL</i> . The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official documentation.
	Trigger	Create a trigger <i>PostgreSQL</i> . It gives a dialog box for entering the necessary data and setting properties. It should be borne in mind that some of the data may be entered unless there are related items in the project, for example, you can define the reference table of trigger only if there is at least one table in the project, in this case the combo contain references to this table allowing the hook to trigger. The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official documentation.
	View	Create a view <i>PostgreSQL</i> . It gives a dialog box for entering the necessary data and setting properties. Among the functions <i>pgDesigner</i> is also a wizard to help you create custom views. The item is part of the objects under <i>PostgreSQL</i> , and the description will therefore refer to the official documentation.
	View Wizard	Run the wizard to build a view. It gives a dialog box where you can select the tables and fields components view now is not managed the definition of WHERE conditions. The object can be created manually amended. At the end of including property, will create a new vision and added to the project and in the diagram.
	Chart Update	Runs updating the graph. This feature redraw all objects in the graph, correcting any visual anomalies. In some cases, employees also the speed of the processor and

 Reorganize diagram	<p>video card, as well as the number of objects in the diagram, it is possible that the graphics engine is completely updated compared to the settings of objects in the draft, so they may be abnormalities occur purely visual, that you can arrange forcing the application to upgrade the entire diagram.</p> <p>This function performs the repositioning of the objects in the diagram based on a very simple algorithm, placing objects in the diagram according to a purely mathematical, without taking into account any relationship between the objects themselves. The objects are arranged in order of creation, starting from left to right and from top to bottom on the chart; to reaching the limits of the design, repositioning start spreading again coordinated by initials. This logic can be applied initially to the first loading of a project from a database or file sql, to allow users to realize the magnitude and the number of elements of the project, allowing then placing objects in a more humanly logic.</p>
 Find objects	<p>This function opens a dialog, allowing you to search for items within a project. For a detailed description of this feature please refer to the later chapters.</p>



Project	
 Open Projects	<p>This menu is displayed unless there are active projects. It contains a sub-menu, with the list of projects open, each of which in turn contains another sub menu with a list of views of the project, including the Main View. Selecting one of these views, change the appearance of the graphics, and display all objects related to the same view. It should be borne in mind that the selection of a view puts the project in a state of change, because the setting is considered as an amendment and saved in the project file and the subsequent loading of the last project override your view was selected.</p>
 Create Database	<p>This feature allows you to create objects directly on an existing database on a server <i>PostgreSQL</i>. It gives a window for the selection of types of objects you want to create, then displays the window to log in and connect to the database server. It should be borne in mind that you used to access the database must have the appropriate write rights to create objects contained in the draft.</p>
 Export SQL	<p>Create the necessary SQL commands the structure of the project and saves them to a file with extension sql, or display them in an appropriate dialog box. As for export database, there is a window for the selection of types of objects you want to create.</p>
 Print Diagram	<p>Runs printing graph of the project. It gives a dialog box that displays the graph divided into pages, as configured in the parameters of printer; pages can be esporate as image files or sent directly to the printer.</p>
 Print Report	<p>Through this item on to a further sub-menu containing some items with which you can create reporting project in different formats.</p>
 HTML Format	<p>Run report of the project, containing all the information concerning the project and object, in html format. The report is then displayed on a special dialog box, from which you can then print or save the contents on file.</p>
 PDF Format	<p>Run report of the project, containing all the information concerning the project and object, in pdf format. The report is then displayed on a special dialog box, from which you can then print or save the contents on file.</p>

 TXT Format	Run report of the project, containing all the information concerning the project and objects, text format. The report is then displayed on a special dialog box, from which you can then print or save the contents on file.
 Project Configuration	Through this item allows access to the general settings of the current project. The function presents a dialog box that lets you change the parameters of the project, including the name and version of the database. It should be borne in mind that the program takes into account the characteristics of <i>PostgreSQL</i> versions of choice, including the size of names and the existence of certain types of objects (which could be available in later), so it is possible that some entries the main menu and some of the buttons in the instrument panel are disabled.
 Project Statistics	View a window to the general state of the project, the total number of objects separated by type, a list of objects and fields of the tables sorted by name, links and other information. Through this window, you can change access on individual items contained in the draft.
 Display Set	To define views of project globally. In the window that appears management is possible to create / modify / delete views (except the Main View), add or subtract elements.

Tool

 View log	During procedures elaborative and management within <i>pgDesigner</i> are issued messages on the current status of the compilation, especially in the case of error or warning, these messages are stored on a file that can be watched to see if any abnormalities. The feature on the menu allows you to read this log file, the content of which is shown in text format on a dialog box; through this window you can send the contents to file or printer.
 Options	With this option, you can access the settings general <i>pgDesigner</i> . This opens a dialog box where you can set some basic characteristics of the application and affect the modalities for the establishment of new projects, it is possible to vary the appearance of open projects, such as color or detail with which items are displayed. It should be borne in mind that the amendment of some of these settings will override the customizations made on some items or property projects open.

Help

 Help	Show window containing this help.
 About pgDesigner	Displays a dialog with general information on the application.

Popup Menus

As mentioned above, during the various phases of project management, you can access some accessories menu, depending on the type of operation that can be made at this time. These menus can be activated at the click of the right mouse button in an area of the diagram, in particular on an object or objects on the right of this board of the project, next to one of the names on the list. These menu enhancements, commonly called popup menus allow access to some features of the program, however, also present in the main menu.

The Menu Project

This popup menu is displayed during operations in general about the project, such as saving, printing or updating of the diagram. The menu is activated with the right mouse button next to the board tab on the project or name tops the list of objects on the right side of the card. The options shown (or voices) correspond to those present in the main menu under "Project", and likewise the general menu, some of these can be disabled depending on the current status, or submit additional evidence.

The Menu Edit

















This popup menu is displayed during operations performed on a particular object or to create a new object on the record. The menu is activated with the right mouse button next to an object on a diagram or names listed items on the right side of this card project. The options shown (or voices) correspond to those present in the main menu under "Edit" and likewise the general menu, some of these can be disabled depending on the current status, or submit additional evidence.



The Menu Bar

Under the main menu is a horizontal panel, which contains a number of buttons and some combo (see figure).



Using the buttons on this panel can be accessed quickly to some of the features present in the main menu, in particular:

- | | | |
|---|------------------------------|--|
|  | New Project | Create an empty project. See the description in the main menu. |
|  | Project Open | Get a file previously saved project. See the description in the main menu. |
|  | Import from database | Import structure from an existing database, hence creating a new project. See the description in the main menu. |
|  | Save Project | Save the current project files. See the description in the main menu. |
|  | Save Project as... | Allows saving project on file with a different name and different format. See the description in the main menu. |
|  | Print setup... | It allows you to configure the parameters of the press and the default printer to send prints. See the description in the main menu. |
|  | Diagram Update | Runs updating the graph. See the description in the main menu. |
|  | Reorganize diagram | This function performs the repositioning of the objects in the diagram. See the description in the main menu. |
|  | Create Database | This feature allows you to create objects directly on an existing database on a server <i>PostgreSQL</i> . See the description in the main menu. |
|  | Export SQL | Create the necessary SQL commands the structure of the project. See the description in the main menu. |
|  | Diagram Press | Runs printing graph of the project. See the description in the main menu. |
|  | Print HTML Report | Run report of the project, in html format. See the description in the main menu. |
|  | PDF Print Report | Run report of the project in pdf format. See the description in the main menu. |
|  | TXT Print Report | Run report of the project in text format. See the description in the main menu. |
|  | Project Configuration | Through this item allows access to the general settings of the current project. See the description in the main menu. |
|  | Project Statistics | View a window to the general state of the project. See the description in the main |

- menu.
-  Display Set To define views of project globally. See the description in the main menu.
 -  Options With this option, you can access the settings general *pgDesigner*. See the description in the main menu.

In addition to the buttons, there are two combo, which you can select the current project (between projects open) or the current display (including displays on the project).

The panel can be made invisible by acting on the general options.

The Panel Tools



The instrument panel is located to the left of the window, and contains buttons necessary to the creation of objects in a project. Each type of object is identified by an icon different. Currently, the scheme has the following items: areas, stamp, domains, procedures, relations (foreign-key) sequences, tables, tablespace, text, triggers, views and types, some of these objects, such as areas and texts, will be used solely for the look and feel of the plot, failing to function on the management of the database.

The panel has been implemented for convenience, given that the same features are also present in the main menu, and can be made invisible by acting options in general *pgDesigner*.

Please refer to the subsequent chapters for a detailed description of the individual objects.

The Status Panel

The panel of state, contains all the information about the status of the current project, in some cases may have features activated via mouse click in certain areas of the panel. The area is located at the bottom of the window and is divided into sectors, each designed to provide a specific information (from left to right):



LED:	If yellow, which indicates is gearing creating a relation.
Definition	Selecting an object report from the Main Menu or the Panel Instruments, active mode to
Relation	create a relationship between two tables. This process begins by selecting (with the click of the left mouse button) a table in this diagram, and ending with selection of the second table (with the click of the left mouse button) that you want to relate. The mode is abortive if the selection is made on an object other than a table or clicking on a blank area of the diagram.
LED:	If yellow, which indicates an object is gearing moving.
Object	This indicator is illuminated when an object is in drag, or when an object in this drawing is
Movement	selected with the mouse and moved on graphics. The release of this mode disables and the indicator is turned off.
	It should be remembered that the dragging of an object puts the project in a state of altered to allow storage of new coordinates of the object, so they will be asked to rescue the closure.
LED:	If yellow, which indicates an object is gearing changing.
Object edit	The indicator is activated when it is in edit mode, or when you create a new object, or selected with a double click or graphics directly in the list on the panel on the right, in which case there is a window to allow the changing the properties of the selected object. At the closing of the window to amend the indicator is off.
LED:	If red, indicating that the project has been changed and the changes have not yet been
Amended Draft	saved. To save the project the indicator is off.
File extension project	This field contains the extension of the current project files (default = ini).
Location of the current object	This field contains the coordinates x, y the upper left corner of the current object, as is currently positioned in the diagram and the current display. Double click with the left mouse button on this field, it displays a window in which you can manually change the current coordinates of the object, the function is similar to the movement done with the mouse, except that the coordinates are, in this case, set manually.
	It should be remembered that the movement of an object puts the project in a state of altered to allow storage of new coordinates of the object, so they will be asked to rescue

the closure.

Printer status The field shows the current settings of the press, such as size, orientation, resolution.

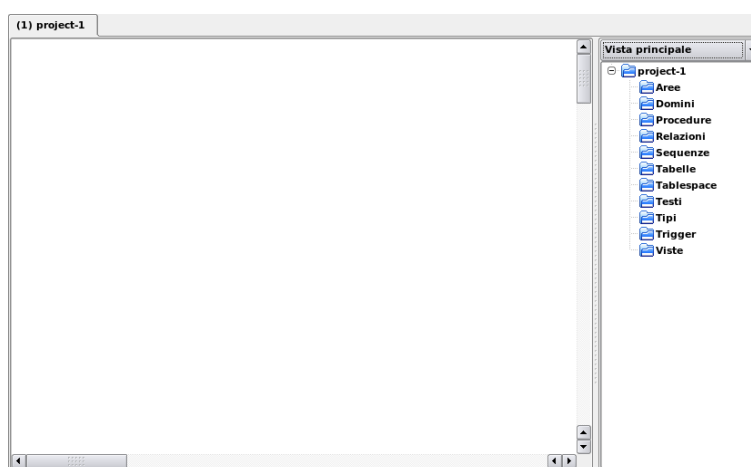
Type and name In this area displays the information on the type and name of the currently selected. If
of the current there are no objects selected, the field is empty.

object

The Panel Projects

The Panel Projects is located at the center of the main window, and in the absence of projects is hidden. At the opening or creating a new draft of this panel is made visible; any open project is associated with a folder, whose label will contain the name of the project.

To make visible, and so on, a project simply select the folder by clicking on the name of the tongue or through the main menu.



Area of Design / Diagram

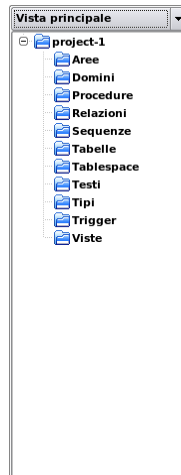
Each folder contains a project that is designed graphically diagram of the project, while on the right is a list containing visible, broken down by type, the objects of the project on this list is a combo with which you can select the view (or display) active. The area of the diagram has a total size of about 4x3 sheets of printing A4; through the scroll bars on the right and bottom, you can move to the diagram, and then reach any objects and out of view.

The View List

On the right of the diagram, as I said, this is an area containing a combo and a list (similar to a directory), the combo contains a list of the displays in the project, and with which you can select one and make it visible in diagram. The name appears view shows the active current, and the selection of a different view will change from the appearance of the diagram, displaying only objects belonging to that view. Each project has a default view, which can not be changed or erasable and that contains and displays all objects of the project, this view shows the label "Main View."

The Object List

Under the combo above, we have a list that shows a structure folders, similarly to a directory system, with groupings of objects by type in the project, each of these folders are listed individual objects, represented by its name (unique), and by a group. Selecting an object here, this becomes the object or subject is active, double click on the name of an object active mode data of the object itself, and the opening of the window on property management.



Project

To enter items in the plan is necessary to create a new project, a load from an existing file, or import directly from an existing database. Each project can be saved to a file and reloaded at will. Currently there will be two file formats: ini and xml, and the two types of files differ only in the syntax, both containing the same information, for which a project can be stored independently on a file type or the other, without any loss of information.

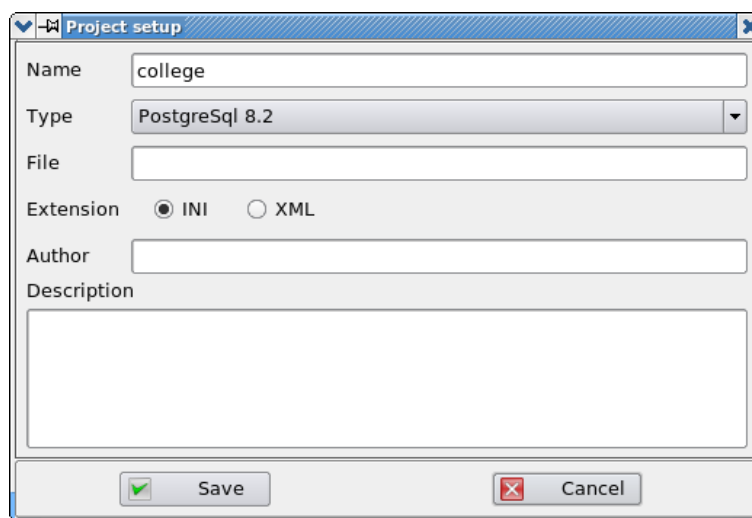
The project files are stored all settings of the project (eg name and version of the database, personalization, etc..), The properties of objects and their position in the diagram, taking into account differences between different display.

In the following pages describes the basic functions for the management of the projects, while the file structure of the project is described in subsequent chapters.

Create a New Project

To create a new project simply select the item within the group File Main Menu. It gives a dialog, which can be set properties of the project, including the name and version of *PostgreSQL* you want to use or where to direct the project.

As shown in the figure, in addition to name and combos to the list of drivers *PostgreSQL* anticipated *pgDesigner*, it is possible to determine in advance the scope of the new project file, enter the name of the author and a brief description of the project itself. Except for the name and type of database, all remaining fields are optional, or will be required when necessary.

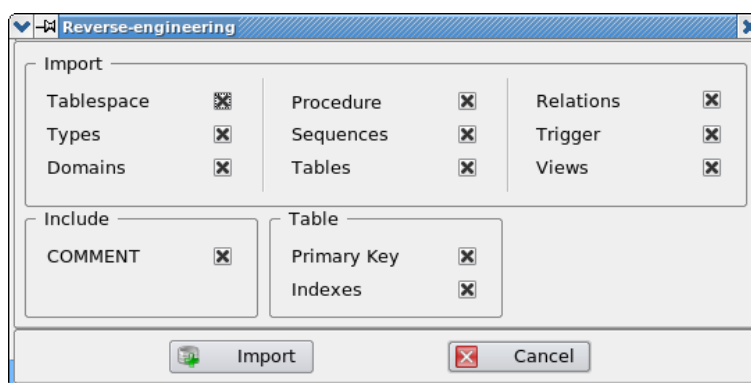


In the event of cancellation under development, the project will be eliminated.

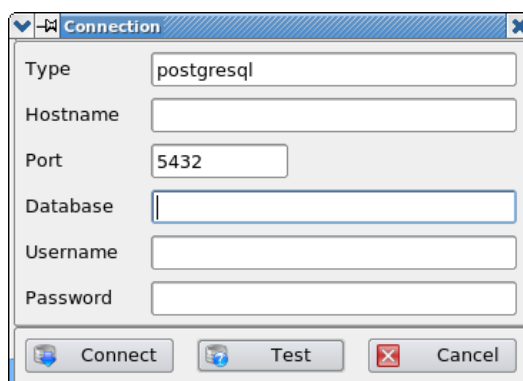
Create a Project importing structure from a database (reverse-engineering)

Selecting the item under the file menu, you can start the process of loading the structure from an existing database, this *PostgreSQL* on a server (local or remote). Before you make the connection, there is a window in which to determine which types of imported items. The feature loads all the objects related to which the connection is made, discarding all other elements even if present on the same database (eg system objects). Depending on the version of the *PostgreSQL* server in use, it is possible that some items are not available, regardless of the selection made in the settings loading.

The figure below shows the dialog box for setting the parameters of loading:



The login window allows you to specify the data required for connecting to the database server. Except password, all settings are saved in the configuration file of the program, and restored the next access. As shown in the figure below, the window provides the opportunity to enter all the parameters useful for the connection. According to the configuration of the server, some of these data can not be necessary or defer the Classic *PostgreSQL*. Through the Test button is unable to verify the correctness of input parameters, while the Connect button connects and start the process of loading. The Cancel button allows you to cancel the procedure, in this case the project is eliminated.



During loading from the database will see a progress bar, which will show the progress in real time. Once started, it is not possible to cancel loading, it is necessary to wait until the end of the process.

Opening an existing project

pgDesigner allows restore a project, uploading relevant information by specific structured files, which contain all the necessary information. Conceptually files are text files, and are set to include all the properties of the project and objects contained therein, including all customizations level display, color and position in the diagram. At present, there are two types of load, described below, essentially based on files in the system.

File Loading

Also through on this item in the Main Menu under the File, you can load a previously saved project files. It gives a dialog box that lets you select the file to load from disk based on its extension (ini or xml). Once uploaded, the file extension is set as the default project and used in the subsequent rescue.

During the upload of the file and the decoding of its content, you will see a progress bar, which will show the progress in real time.

Uploading files from Sql

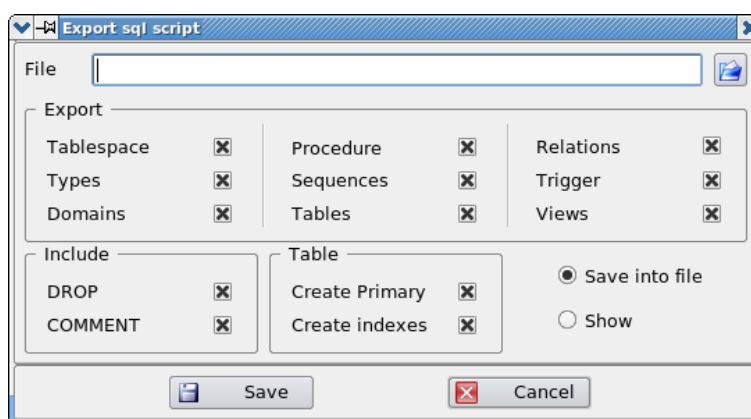
In addition to standard project files, you can also create a project uploading a file containing commands purely SQL (eg export directly from a database). It should be borne in mind that the order of the objects in the file, and then the relevant SQL, must respect the order of creation of the objects on the database, if not the function of loading might be unable to make the necessary connections and properly defining relations between objects. For the creation will be taken into consideration only the only items under *pgDesigner*, and will be excluded from loading all those who are not recognized. From bear in mind that the complexity of decoding SQL commands may cause errors and thus canceling loading resulting in cancellation of the new project.

As for the standard project files, while uploading files and the decoding of its content, you will see a progress bar, which will show the progress in real time.

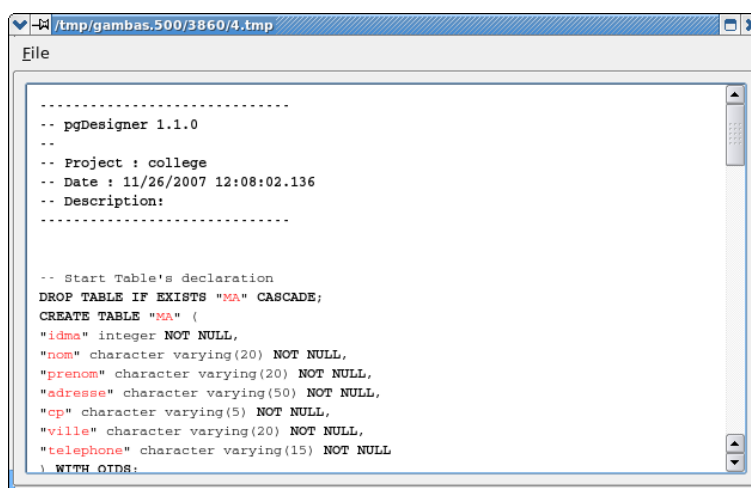
Export Project

You can export a project, or SQL related objects contained therein. In Group Project of the Main Menu is its voice. This opens a dialog box that, similarly to upload feature from the database, which allows you to set export items and some additional features, as shown in the figure below, the window shows a top field in which you can type the name file (including path) on which you write the SQL commands. Optionally, by the icon located on the right side of the field, you can go right on the disk directory in which to save the file itself.

As already noted several times, certification or less of the types of accommodation provided in the present and for selecting parameters export depend entirely on *PostgreSQL* version selected for the project, it is possible that some of the items are not available and options are disabled.



As an alternative to writing on the file, you can export and see the sequence of instructions directly to a dialog box, so you can immediately see the final result, from this window will be possible then save or print the contents .



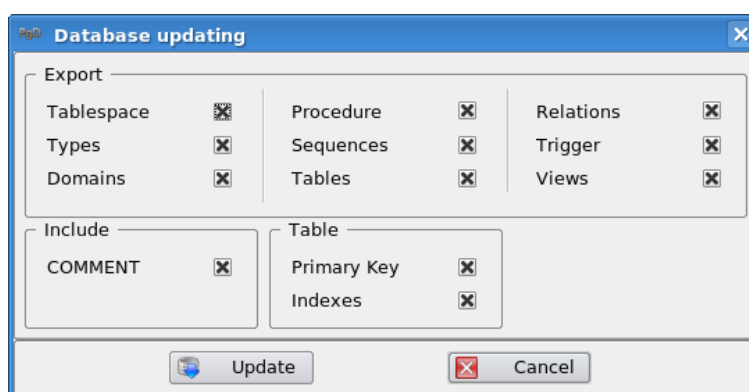
Exports to file Sql

As mentioned in the previous paragraph, you can create and then export to file SQL commands relating to the structure of the project. The resulting file is a text file, with sql extension. After setting the parameters defined and export the file name to be written, will start the procedure coding SQL, during which you will see a window with a scroll bar indicating the progression of processing. Once started, the process can not be stopped.

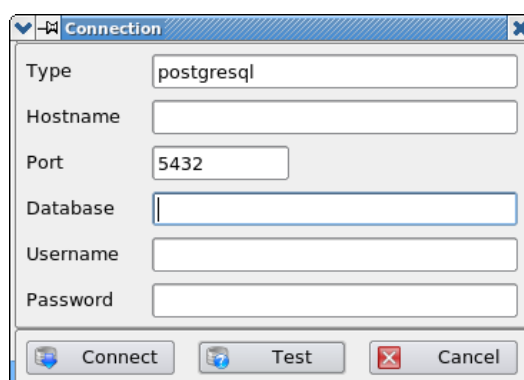
Exports on Database

Over extraction of SQL commands on files, *pgDesigner* also provides for the possibility of exporting the project directly creating the database on a server *PostgreSQL*. As for export to file, the procedure for export database provides a window on which set the parameters and objects you want to create.

As already noted several times, certification or less of the types of accommodation provided in the present and for selecting parameters export depend entirely on *PostgreSQL* version selected for the project, it is possible that some of the items are not available and options are disabled.



At the end of the definition of parameters, will be presented the window to connect to the server *PostgreSQL*.



It should be borne in mind that the operation is carried out with the name and password that is logged, so it makes sense that the database exists and that the user is the owner, or at least has all the rights necessary

for the creation of objects contained in the draft. Items already in the database, owned by other users, and corresponding to those contained in the draft cause blockage of the procedure, the release of an error message.

It is important to note that, at present, it is not possible to perform an upgrade of existing objects in the database, any existing object corresponding to that contained in the draft, will be completely overwritten, resulting in elimination of all existing data. Before performing the procedure for export, we recommend making a backup of preventive and comprehensive database.

Note: the possibility of creating database in a particular type of object is dependent on the server version PostgreSQL, it is necessary to be careful before proceeding, and any inconsistencies cause blockage of the procedure and release of an error message.

During the process of creating the database will see a progress bar, which will show the progress in real time.

Saving Project

To save the project file, so that you can restore later to make further changes or create a database. As mentioned previously, you can save files on the project in two formats: ini and xml. It gives a dialog box to allow the type of the file and its extension (ini or xml), the extension of the file will be used as a default in the draft and used for the next rescue.

According to the selected file type, the time required for rescue can be very different; usually an ini is faster while, by its inherent structure, saving a file xml be much longer, the same data.

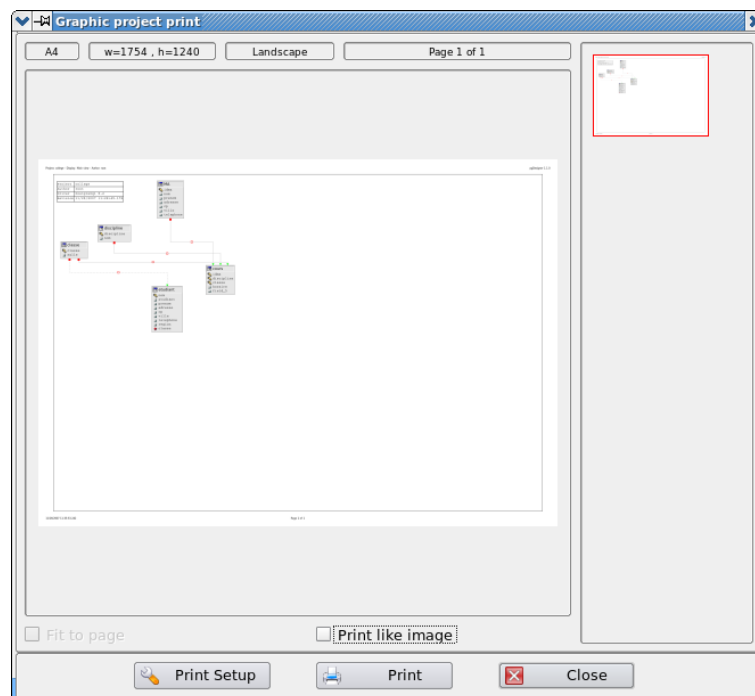
When you save the file you will see a progress bar, which will show the progress in real time.

Printing the Diagram

Compared to the previous version of *pgDesigner*, the preview and print diagram of a project has been greatly enhanced. With this feature you can send graphic diagram on the printer or a disk image file.

Preview

By selecting the item in this group Project of the main menu opens a window, as shown below:



As shown in the figure, the preview window is made up of some areas. In the upper zone are some fields that display the current configuration of the press, which you can change using the "Print Setup". On the right is a vertical panel, which displays thumbnails of pages in which it broke down the plot of the project, depending on size, their orientation and print resolution, the format of these miniatures may be different. At the center is a panel that displays the current page, the page design is for obvious reasons dimensioned to fit to available but gives an idea of how the release will be made final.

A page can be selected and displayed in the center panel by clicking on the left button on a thumbnail displayed on the side panel, the selected page will be designed with dimensions greater central panel, while the miniature will be highlighted by a border of red. In one of the fields to the upper window, see the current page number and the total number of pages which make up the plot.

Given the size of the plot of the project, as described in previous chapters, *pgDesigner* provides an algorithm that determines the areas useful to the press, eliminating areas that do not contain objects or graphic elements (see areas and key relation); this, so avoid unnecessary printing blank pages and consequently

wasting paper (a little economy and less wood to be cut).

As mentioned, each page contains a section diagram of the project to identify pages on each sheet is drawn one edge and, externally, basic information, including the name of the project and the date of revision, the date press and page number.

As mentioned, you can at any time change the print settings, accessing to a window in which you can determine both the driver and the destination printer, the size and orientation of the prints. The setting will remain valid even after the closure of the preview window, and reported on the diagram.

As you can see in the window are two check, the first everything possible to reduce the graph in a single page. Obviously, the size of the graph can heavily influence this choice, a very large graph could be illegible if reduced and printed on a single page, specially trained in very small print. The second check can save the graphic image to a file.

Print

As described more fully enough in the previous paragraph, using the preview window can be sent to the printer pages which make up the diagram, with the print settings defined in Print Setup. It should be borne in mind that the entire plot of a project is built on a scale indicative of 4x3 A4 pages, so as mentioned in the previous paragraph, it is possible that if I check "Fit to Page" selected the final print is illegible.

Image File Creation

Check "Print as Image", on the window, allows you to export the graph image to a file, so that we can enter into any documentation. Currently managed a number of image types: PNG, BMP, GIF, JPG and XPM.

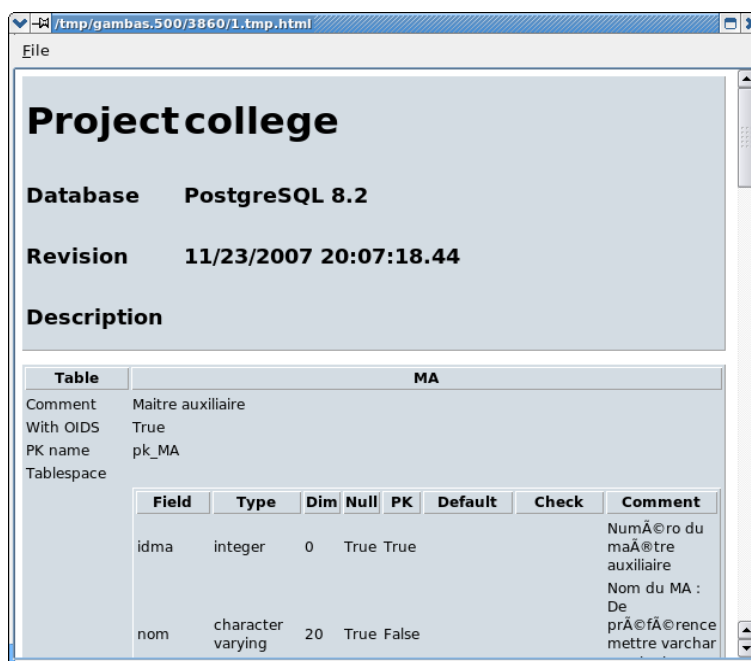
Reporting

Among the various backgrounds provided in *pgDesigner*, it is possible to produce a series of draft documents in different formats. The documents contain, in a textual and descriptive content of the project, the objects contained therein each with its respective properties and configurations. At present, and the inclusion of a new library, it is possible to produce the documents in three different formats: html, pdf and text. The result is displayed on a window where you can export the contents to file or send it directly to the press. In the following paragraphs describes the particularities of each of these formats.

Documentation in HTML format

Selecting its voice in the group Project of the main menu, you can create project documentation in HTML format, which makes it possible to be read in a web. When the structure of this type of documentation is fixed, with color graphics and default settings. As with other formats, documentation mainly consists of a header, listing all the general properties of the project, and the details of each object with all its features, except for the property within the graphic diagram .

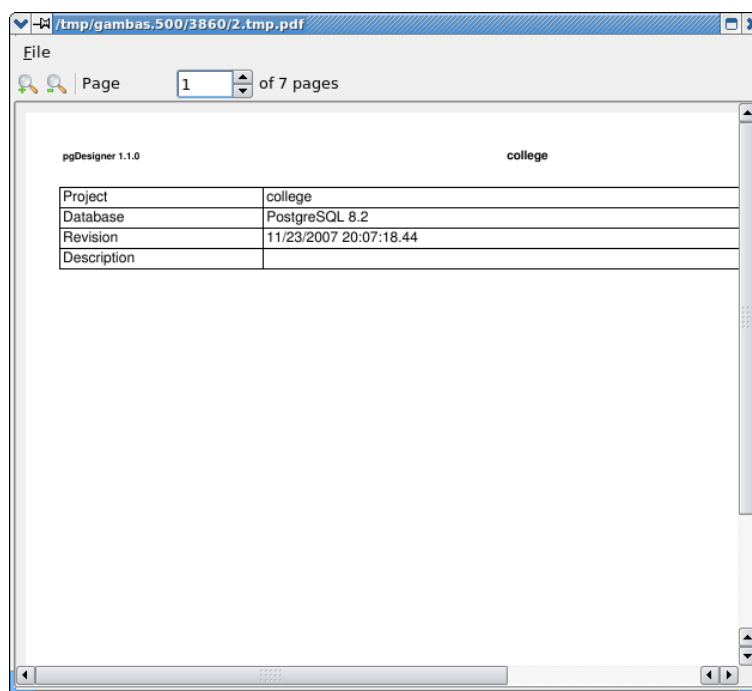
The window shown in the following figure shows an example of documents in this format. As you can see, the window also presents a simple menu, through which you can save the contents to a file or send it directly to the press.



Documentation in PDF format

Selecting its voice in the group Project of the main menu, you can create project documentation in PDF format. As with other formats, documentation mainly consists of a header, listing all the general properties of the project, and the details of each object with all its features, except for the property within the graphic diagram .

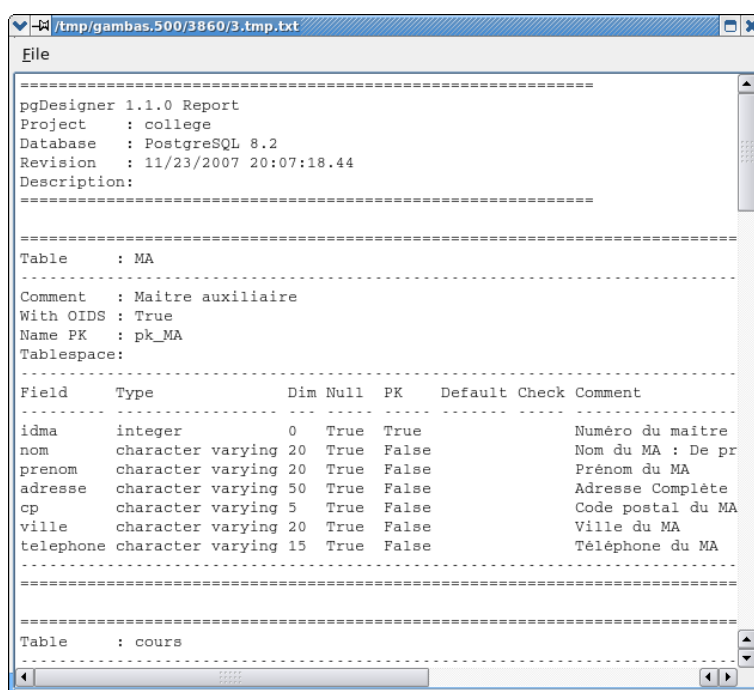
The window shown in the following figure shows an example of documents in this format. As you can see, the window also presents a simple menu, through which you can save the contents to a file or send it directly to the press. Under the menu is a bar that allows you to view pages components documentation, size and the content according to their needs.



Documentation in TXT format

Selecting its voice in the group Project of the main menu, you can create project documentation in text format. As with other formats, documentation mainly consists of a header, listing all the general properties of the project, and the details of each object with all its features, except for the property within the graphic diagram .

The window shown in the following figure shows an example of documents in this format. As you can see, the window also presents a simple menu, through which you can save the contents to a file or send it directly to the press.



Project View / Display

For each project, you can create views (display, or even schemes), which is given a particular mode of visual objects of the project.

Main View

Each project is linked a Main Vista, which contains all objects of the project. Through the creation of additional views, it is possible to separate objects in logical groups. An example could be seen in a group "Codes" all purely objects containing data encryption (eg means of payment), or a sight "customers" all tables on the management of customers (eg data, invoices, etc..). Unlike personal views, the Main View can not be altered or erased, since it contains all objects of the project.

Creating Custom Views

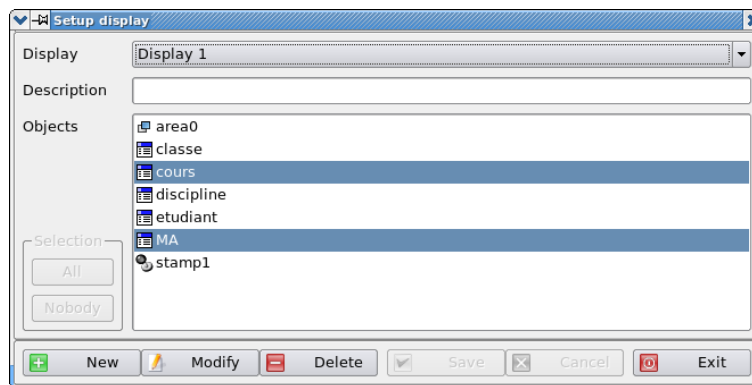
As described in the previous paragraph, except for the main view, all views or custom created can be modified (adding or removing items), or removed from the project. Any transaction made on the view does not affect the objects of the project, but it is also possible to manage every single item regardless of its place in any view, in this way, any change in configuration of an object will be reflected in all views containing it . Unique setting employee sight, is the position of the object in view, which can be different and modified independently of the other, so it is possible to modify the layout of each view, independently, in different ways by placing objects on it compared to other views.

Managing objects in a customized view

It should be noted that, with regard to the objects table view will be reflected in all the relations between the same table will not, however, appear linked to relations tables Not in sight, although one of the tables is present is then essential that to see a relation must be seen in all and two related tables.

Note that the relation objects are the only items that can not be defined manually within a sight, and therefore depend on logical conditions.

Through the window (selectable from the main menu), as shown in the figure below, you can create new views or amend those already in the project.



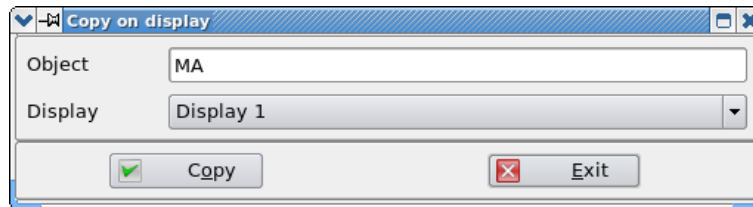
In combo "Display" lists all views on the project with the exception of Main View, while the list "items" displays a list of all objects in the project. For each view, you can enter a brief description.

At the bottom of the screen are buttons with which you can create a new view, modify or delete an existing one, the buttons "Save" and "Cancel" will be shown exclusively in the state of change, respectively, and allow you to save changes made a view or restore the previous conditions. Any change will be reflected immediately in the graph at the close of the window.

The name of each new display is set automatically by the program, according to the order of creation. On loading a project, all displays will be renamed in how were saved.

Selecting a display in the combo, all objects in view associated are highlighted. It is possible to modify the list of items vision, selecting with the mouse; removing the highlight of an object, it is removed from display. Selecting multiple objects can be done by holding down the control key keyboard and acting with the left mouse click on the elements of the list.

It should be remembered, as well described in previous chapters, and through the popup menu or the main menu allows you to copy an object in a different view, in which case the procedure presents a window in which to define the view of destination :



As shown in figure is reported by the name of the copy, while the combo "Display" presents a list of all views in the project, excluding Main Vista.

Entities and Objects

Each element in a project is treated as an object defined by its properties and characteristics, which will be useful in building the physical element in the database. These properties are saved in the project file along with the position on the charts, including the position within each view defined in the project. The next reading project file, restore the properties and positions previously saved, including their positions in the graph.

Every single object, both visually and with the exception of objects of type relation, text, and areas is shown on the chart as a small window, complete with title (name of the object), and any other details (eg fields a table). A special icon to the left of this title, visually identifies the type (or group) of the object.

In addition to the icon of the group, in order to distinguish one object from another, each element must necessarily have a name, which has unique within the same project. The name is being set creation, but can be changed later (obviously respecting the logic of mandatory and uniqueness). This logic is valid also for *PostgreSQL*, which allows for more objects with the same name, although of a different nature.

As for the objects of type tables and views, you can make visible on the chart also their fields. The tables are displayed: name of the field, type and size as well, if in an index, the index type (single, primary key, foreign key ...). The views display only names of the columns. On the right of each field, you see an icon that identifies whether this is a primary key, a foreign-key or a normal field.

It is possible to set the details of the content of items affecting the look and feel within the diagram, for example tables and views, you can see whether or not the fields.

As for the objects of type relation, for obvious reasons these objects follow a completely different logic; their graphic representation is dependent on the presence and position of related tables. The relation appears in the form of a line, which can be continuous or dashed, according to the type of connection (mandatory or otherwise) between the two tables. At the moment it is not possible to manually move a relation, whose design is done automatically by the program. They are still a few possibilities for customization, internal engine used by the application, which affect some aspects and the visual tracking of a relation: the type of connection and the type of coupling between the two tables. Through the first approach can be defined as the line is drawn from table to table daughter father, which may be only two ways: with a straight or a straight line segments composed of horizontal and vertical perpendicular, which points start and end are linked the two tables. The second approach determines as points of attack on two tables are calculated and shown on the chart: the top and bottom edges of the table or at camps in the relation, in the first case, the line is drawn from the bottom of the table and ending father on the top edge of the table daughter, these two points are represented in real time, at the movement of one of two tables, and their position is calculated automatically even considering the existence of other relations on the same table, in the second case rigging points starting from the right side of the table father at the first camp that makes the relation, and ending on the left side of the table daughter always at the first field in relation.

The point of the line of relation is represented by a small square, while the final point comes as a bit of arrow, this approach will help you understand immediately the direction of relation and how the two tables are linked with each other. As with other objects, you can select active and make a relation with the click of a mouse next to the line or points of attack and if the display is active labels, the click of a mouse on the name of the relation, which is displayed in central line, will allow the selection of relation. Double click on a relation will

bring change for the management of the properties of the object.

Creation

The creation of an object is possible via the Tools Panel, which contains buttons for all types of object that can be included in a project, or through the Main Menu, which contains entries corresponding to panel buttons. In addition to this you can open the popup menu in the diagram, which contains the same items in the Main Menu.

Dimensions

All graphic elements in the diagram, with the exception of objects of type area and relation, are automatically sized, and depending on the type of detail defined in the window of broad options. For objects of type area, you can change their size acting with the mouse at the right and bottom margins, holding down the left button next to these margins and dragging the mouse until it reaches the desired size visually. As already mentioned, the design of relations is performed automatically by the engine, based on the current position of the two related tables.

Movement

All graphic elements, with the exception of objects of type relation can be moved within the diagram, for the movement simply selecting the object with the mouse and holding down the left button, drag the object to the desired position . Double click on the Panel of State at the bottom of the main window, at the field that displays the current coordinates of the object, opens a dialog box where you can set the position of manually entering values in pixels X and Y coordinates corresponding to the upper left corner of the object in relation to the diagram. As already mentioned, the design of relations is performed automatically by the engine, based on the current position of the two related tables.

Editing

As already mentioned in the preceding paragraphs, modification of the properties of an object can be carried out in several ways: with the double click of the mouse on the same within the diagram, or with the double click of the mouse on the corresponding name in the this list of items to the right in the project.

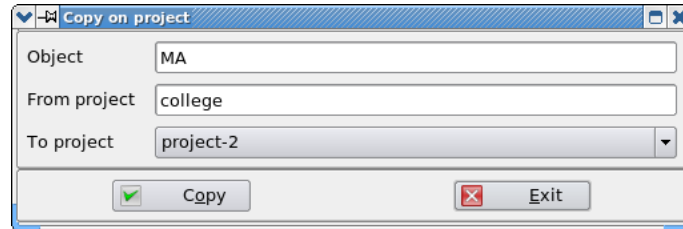
Deleting

You can delete an item from the project by opening the popup menu directly on the diagram of the same at selecting it with the mouse or, as modified, by the list of objects present in the project. Once removed, the object will no longer be recoverable, and all references to it are also deleted, in case of removal of a table, it will also delete all reports related to this.

Copy on another project

As for the copy on display, *pgDesigner* can copy the information and the properties of an object on another

project among the list of projects open. The object in this will be recreated inside the target project, cloning its property and its features, except for references to other objects. If the target project is already an object with the same name, the copy will be canceled.



As you can see from figure in the window a copy appear: the name of the object to copy, the source and a combo containing the list of projects open in *pgDesigner*; via the combo you can select which project will be copied to the selected object.

Description of Objects

This chapter describes in detail all objects used within *pgDesigner*. Moreover be described as changing the properties and characteristics.

Except in certain cases, they will not be given technical explanations on the use and functionality of objects within a database, which can be found in the official documentation of *PostgreSQL*.

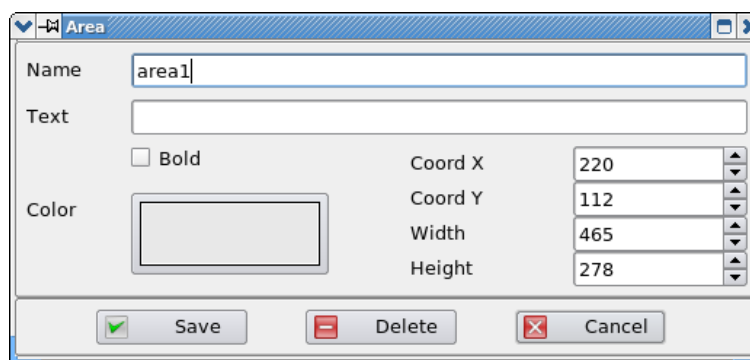
At the current status are provided all the information necessary to create objects from version 7.0 up to 8.2 of *PostgreSQL*, this requires a knowledge of the characteristics of the database that you want to create, so it is possible that some features are not available in the server database that you have or that you use for the project.

By changing the characteristics of each object described in the following paragraphs, there is a dialog, which is different depending on the type of object that is gearing dealing with. All windows are activated described in modal, or you can not act outside of them, until their closure for confirmation or cancellation of operations changes. Normally, each window is made up of an area that has seen the name of the object; a central, possibly divided into folders, containing all the necessary controls for the settings relating to the subject property, and a bar on lower where are the necessary buttons for: saving changes, delete the object or reverse any changes. Any changes made and saved, the data in this window will be stored in the configuration of the object itself and, according to the defined graphics, displayed directly in the graph. For some items, such as tables, there are more buttons to activate the management of individual fields of the object or its indices; for such objects are more windows. It is considered that any changes made and confirmed in the windows to modify elements of an object below, there are void if changes are made void in the primary, for example, tables, changes made and confirmed on individual fields are not annulled by pressing the Cancel button in the edit window of the table.

Area

The areas are a new type of object graph, which allows to draw colored rectangular areas with the opportunity to view a text on the inside, this type of object is useful for instance to define groups of objects in the diagram relating a particular type or function inside the database. These objects have properties not active, or exercise any influence on objects contained in the draft, and can be defined independently of any display of the project. For each area, you can define the fill color and, optionally, a line of text as a description of what you want to enclose. Creating object in selecting the panel instruments, from the main menu or from the popup menu by pressing the left mouse button in a chart displaying and dragging the cursor to the size desired; releasing the button will set the final coordinated. During building, appear on the chart the new area (initially gray color), which constantly monitor the design phase, so as to enable the user to verify in real time the actual size. It should be borne in mind that, for practical reasons, the areas with smaller sides of 50 pixels per side will be automatically eliminated, why not considered useful and to eliminate the possibility of creating invisible objects (such as an object from 1x1 pixels) that could create problems for their cancellation. Then you can change the settings of the new business double-click with the left mouse button on himself, who will present a dialog with the current ownership of the new item. At any time you can move or resize a subject area: in the first case, as with other types of object, just select the object with the mouse and holding down the right button, drag the graphic element to the position desired, in the second case, the size can be changed by dragging the edges or bottom right until he reaches the desired size.

The figure below shows the window is to modify an object Area. Like all objects, the window presents a field where you must enter a unique name in the draft.



The "Text" lets you enter a description, that note will be displayed in the upper left corner inside the graph. Optionally, you can highlight the text in bold, acting on the check.

In correspondence of the "Color" is a button that represents the current background color of the object, pressing this button will access the window for selecting colors, to change the current setting.

In addition to the above settings, you can manually define the position and size of the chart, entering the coordinates corresponding directly at the top left corner of the object and its size width and height.

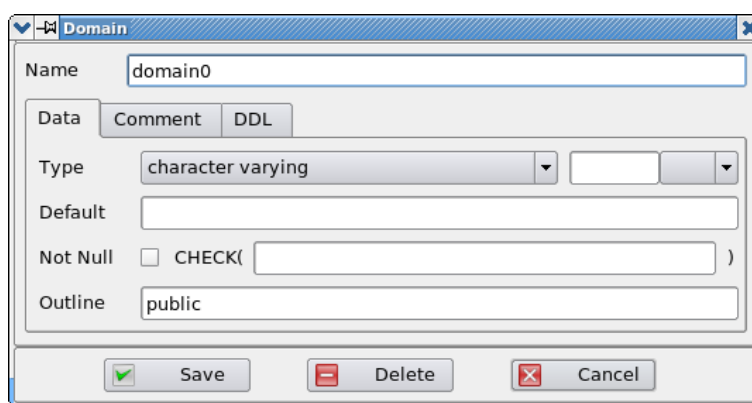
Any such change will be effected reflected in the diagram of the project to close the window change.

Domain

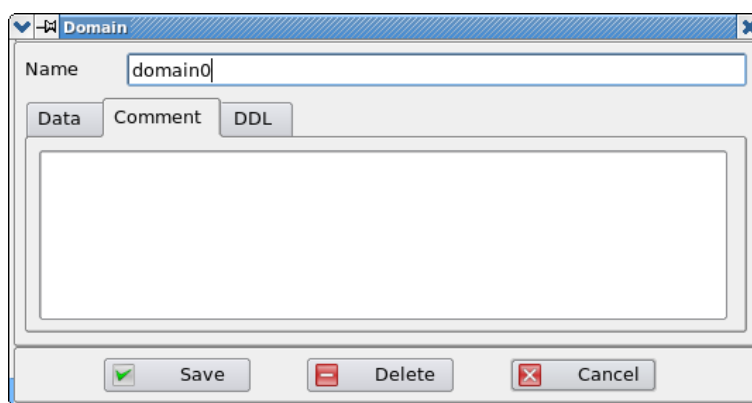
For technical specifications related domains can be found in the official documentation of *PostgreSQL*.

The window shown in the figures below, lets you change the properties of an object domain.

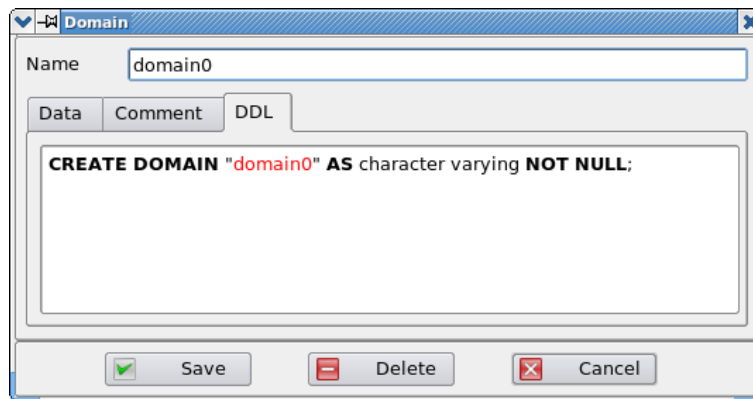
As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project. The window is made up of three folders, grouping in the different logical characteristics of the object



In the first folder "data" are the basic properties of the object. The combo "Type" contains a list of all data types allowed for the release of *PostgreSQL* selected, and if you position your mouse next to the combo, and if it turns tooltip options in general, you see a brief description the type currently selected.



The second folder is an area where you can enter a description of the object; *PostgreSQL* provides for almost all objects a property Comment, to allow to document objects.



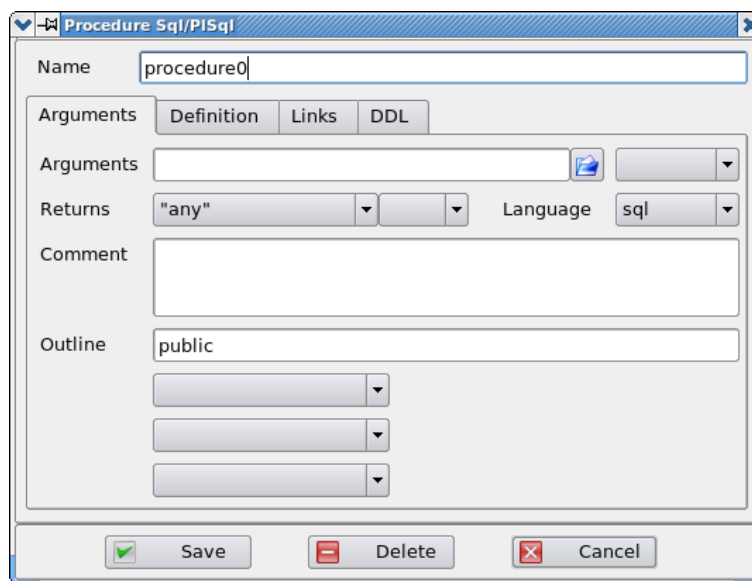
The last folder, exist in almost all the windows on object-oriented database, appear SQL commands on the building and the construction of instruction takes place in real time, and depends on the settings in the property contained in the current window.

Procedure/Function

For technical specifications regarding the procedures can be found in the official documentation of *PostgreSQL*.

The window shown in the figures below, lets you change the properties of an object procedure (or function).

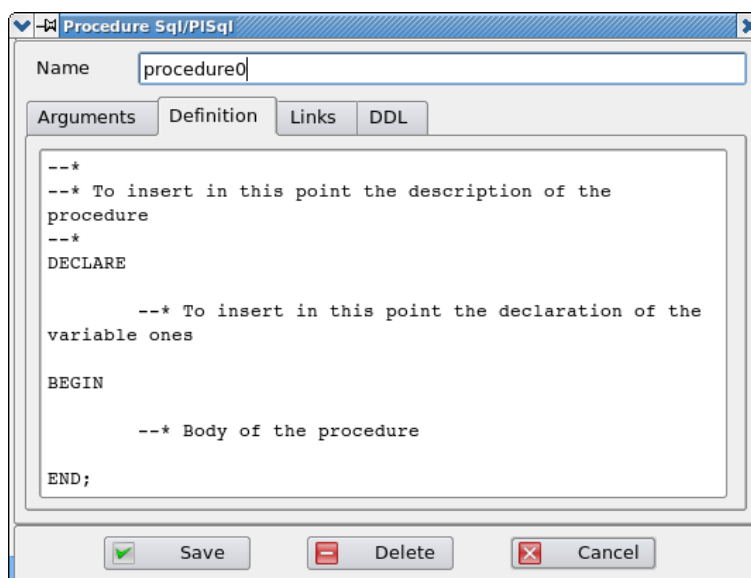
As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project. The window is made up of four folders, grouping in the different logical characteristics of the object.



The "Topics" you can enter a list of parameters of the procedure, separated by a comma; the procedure checks the validity of strings enter, which must match in the types accepted by *PostgreSQL*, including types defined (as described in the following paragraphs). On the right of this field is also a button, which opens a short list containing all types; selection with the double click of the mouse, move the type selected topics in the field.

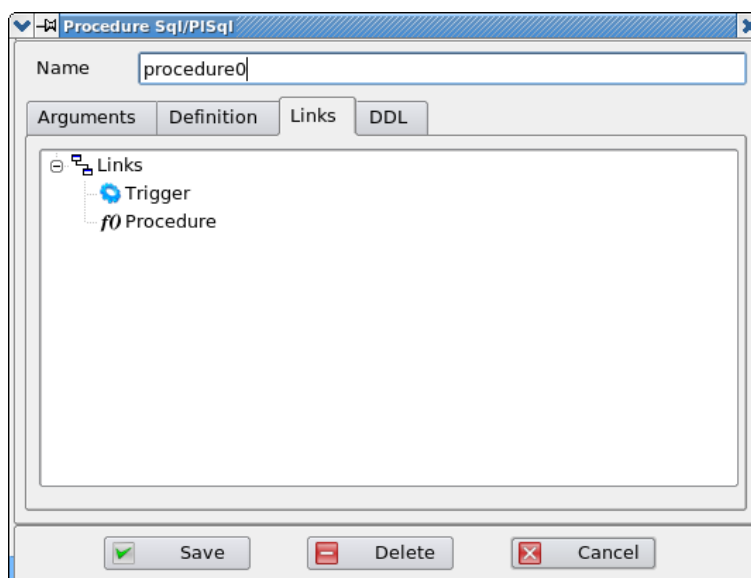
The combo "Returns" contains a list of all the data types are allowed on all versions of *PostgreSQL*, at present there is no distinction between the various versions, and monitoring should be carried out by the user.

The last three combos allow you to define the modus operandi of the procedure.

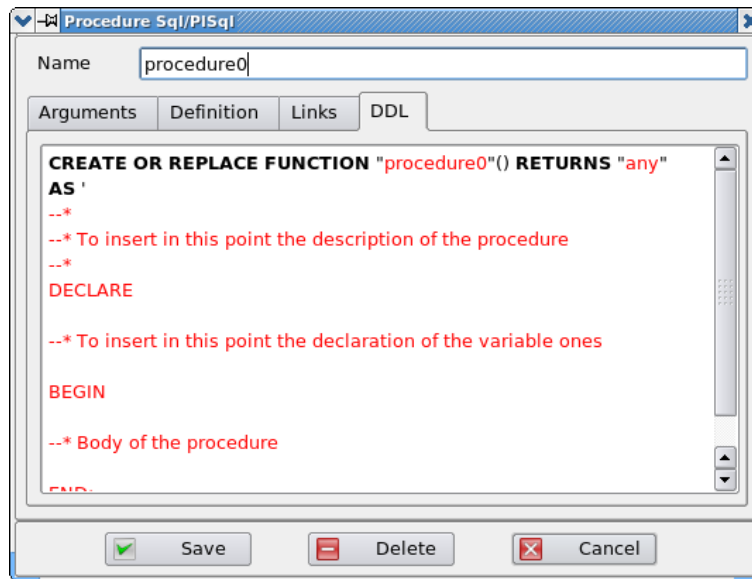


In Folder "Definition" can be inserted in the code sql desired language, according to the approach selected in the previous folder. By default, and only for new procedures, you enter a text already preset, which can serve as the basis for writing function is to be borne in mind that the structure of the text of default Basa language pgsql, it is possible that is not usable if you select a different approach.

Note: Strings included single quotes (SQL syntax), in the definition, are automatically and properly converted for the creation of proper procedure in the database.



The third tab "Links" presents a tree-structured list, which displays any reports of the procedure with the other items contained in the draft. The list is for informational purposes only, and has no function.



The last folder, exist in almost all the windows on object-oriented database, appear SQL commands on the building and the construction of instruction takes place in real time, and depends on the settings in the property contained in the current window.

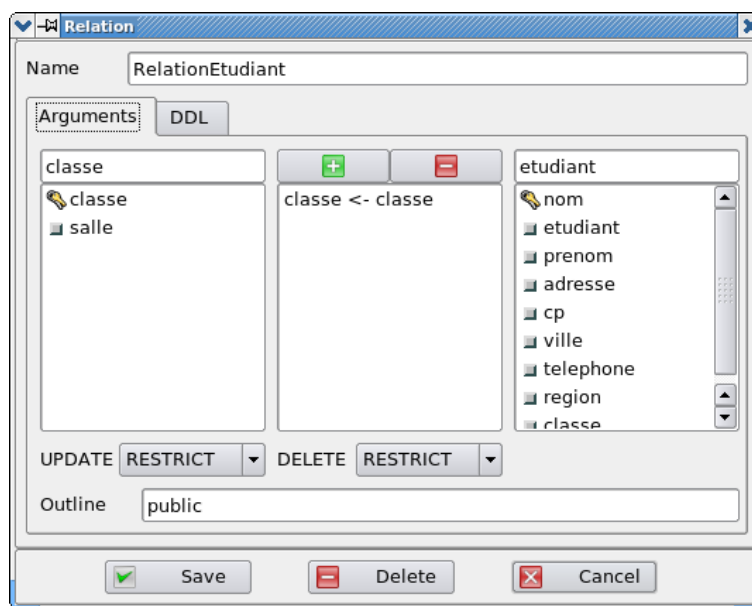
Relation

For technical specifications on relations between tables, see the official documentation of *PostgreSQL*.

Regarding the creation and use of the relations, has already been given full explanation in paragraphs earlier.

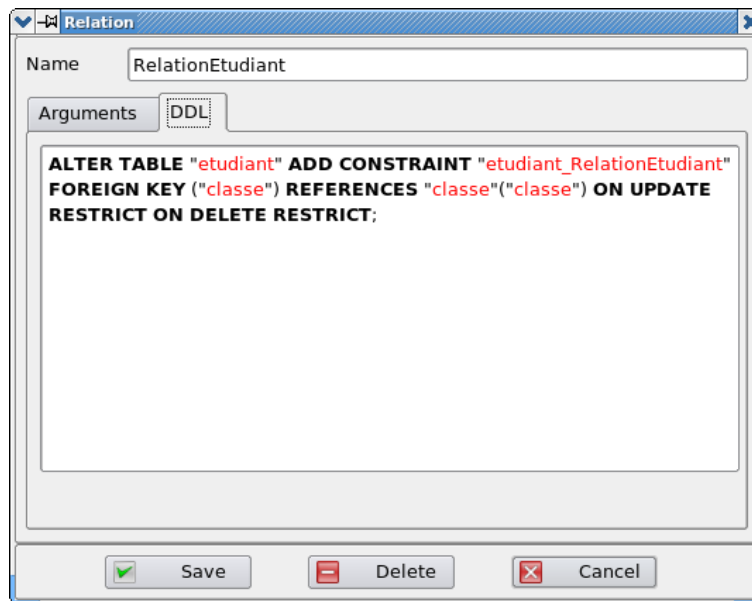
The window shown in the figures below, lets you change the properties of an object relation.

As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project. The window is made up of two folders, grouping in the different logical characteristics of the object.



In Folder "Topics" are three lists. The list left lists all fields in the source table (or father), while the list on the right lists all the fields in the destination table (or daughter), in central list can be found all relations field / field between the two tables. To establish a relation between two tables must be selected in each of these areas to be connected, and press the button at the top left of the central list (identified with a "+"); this field will be selected to populate the list of relations in the central list. It is not possible to create a relation without having at least one item. To delete a row from the list is central to press the button on the upper-right (identified with a "-").

The combo under the three lists are used by *PostgreSQL*, and determine the type of relation between the two tables, and the definition of this type of relation conditions graphical representation of a line linking within the diagram, in order to facilitate the identifying and understanding the type of relation.



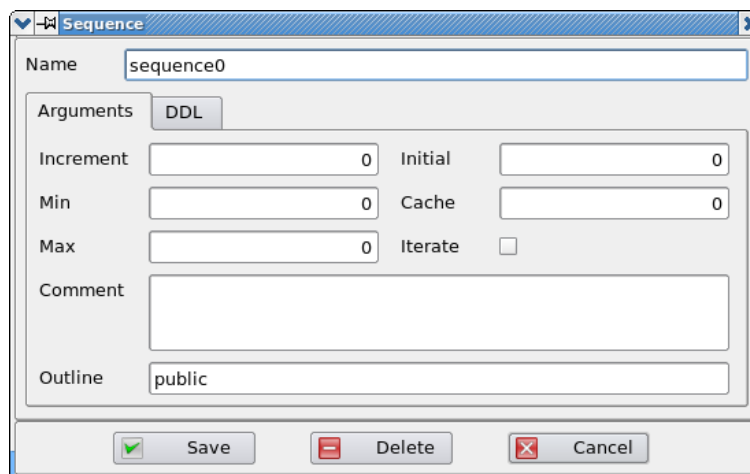
The last folder, exist in almost all the windows on object-oriented database, appear SQL commands on the building and the construction of instruction takes place in real time, and depends on the settings in the property contained in the current window.

Sequence

For technical specifications related sequences can be found in the official documentation of *PostgreSQL*.

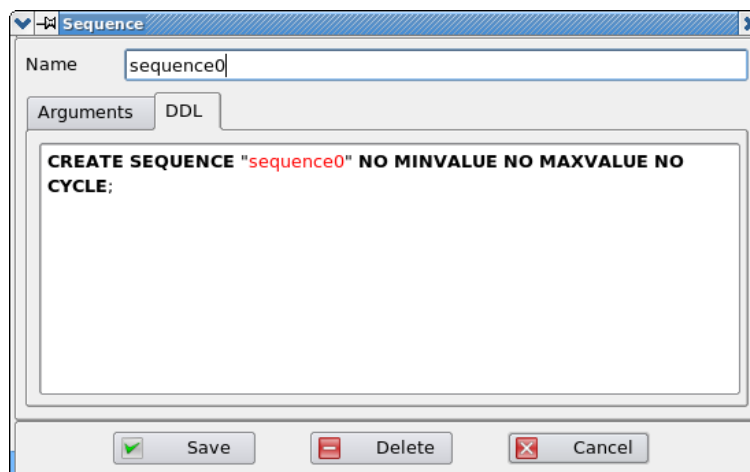
The window shown in the figures below, lets you change the properties of an object sequence.

As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project. The window is made up of two folders, grouping in the different logical characteristics of the object.



The 'Sequence' dialog box is shown with the 'Arguments' tab selected. It contains the following fields and controls:

- Name:** A text field containing 'sequence0'.
- Arguments:** A tabbed interface with 'Arguments' and 'DDL' tabs.
- Increment:** A text field containing '0'.
- Initial:** A text field containing '0'.
- Min:** A text field containing '0'.
- Cache:** A text field containing '0'.
- Max:** A text field containing '0'.
- Iterate:** An unchecked checkbox.
- Comment:** A large text area.
- Outline:** A text field containing 'public'.
- Buttons:** 'Save' (with a green checkmark icon), 'Delete' (with a red minus icon), and 'Cancel' (with a red X icon).



The 'Sequence' dialog box is shown with the 'DDL' tab selected. It displays the SQL command for creating the sequence:

```
CREATE SEQUENCE "sequence0" NO MINVALUE NO MAXVALUE NO CYCLE;
```

The 'Name' field at the top still contains 'sequence0'. The 'Arguments' and 'DDL' tabs are visible at the top. The 'Save', 'Delete', and 'Cancel' buttons are at the bottom.

The last folder, exist in almost all the windows on object-oriented database, appear SQL commands on the building and the construction of instruction takes place in real time, and depends on the settings in the property contained in the current window.

Stamp

The object Stamp is a new object with special characteristics. As for the objects area, the subject has not Stamp functions, both within the project that the creation of the database.

The object is presented graphically with a simple rectangle, which contains information about the project that includes: name of the project, author, version and date database of project review. All this information is gathered from the same project, and then you can not make any changes acting on these objects, and therefore is not a scheduled dialog, and any changes in ownership of the project are automatically reflected on objects Stamp, updating directly their content.

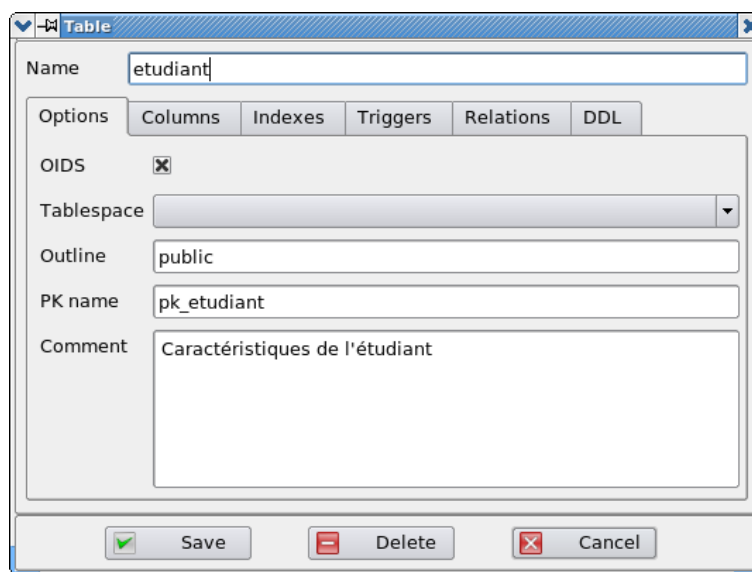
Like all other objects, you can create even more objects of this type in the diagram, and regardless of the display on each project; placement can be performed similarly to other objects, while not absolutely can change the size.

Table

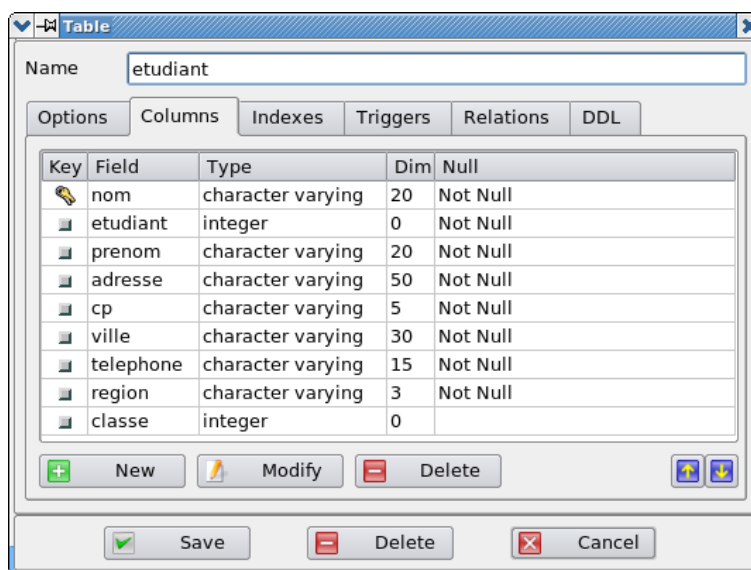
For technical specifications on the tables, see the official documentation of *PostgreSQL*.

The window shown in the figures below, lets you change the properties of an object table.

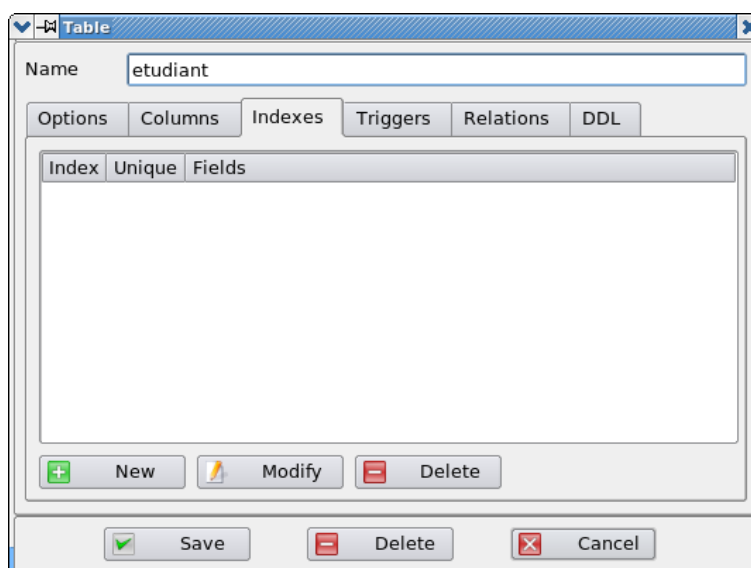
As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project. The window is made up of six folders, grouping in the different logical characteristics of the object.



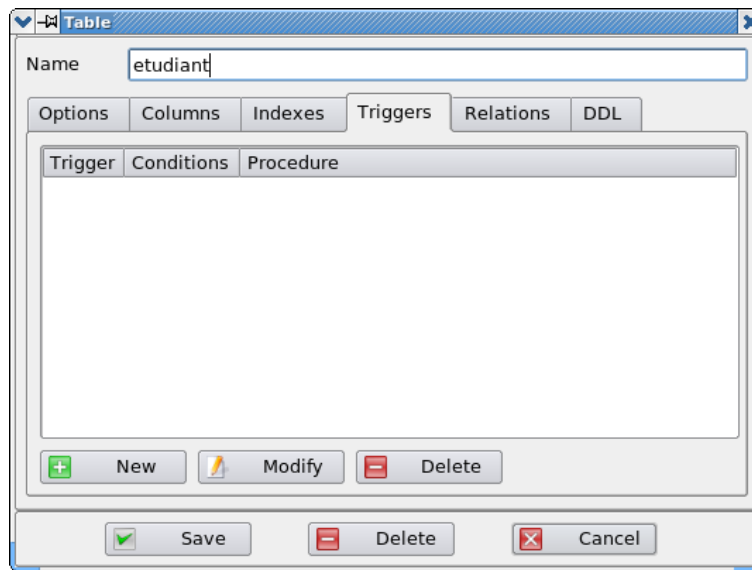
In the first folder "Options" are all properties of the table. The combo "Tablespace" contains a list of all objects tablespace in the project, selecting one, it is docked to the table. It is therefore not possible to give any specific tablespace, in this case, the table will be associated with the tablespace default *PostgreSQL*.



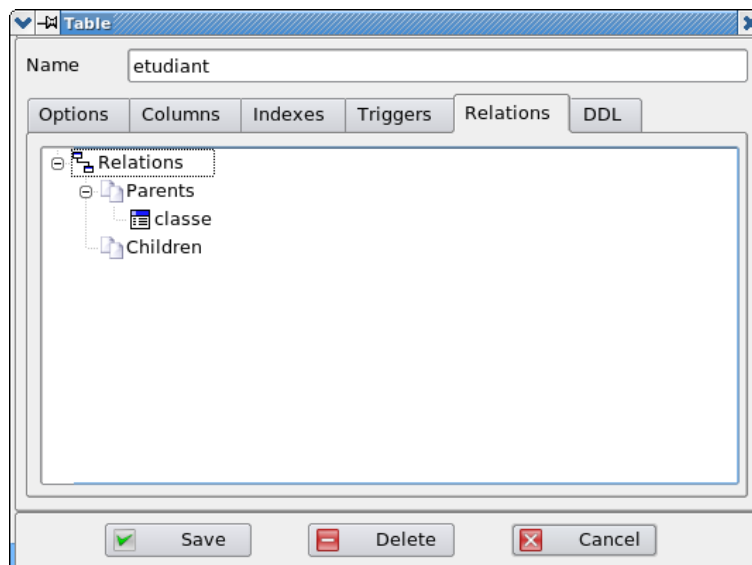
The second folder "Columns" presents a list of all fields and to schedule allows, through buttons appear, access to information on individual fields of the table. Double-click on a row of active list behaves like an edit button, opening the dialog for editing properties of the field.



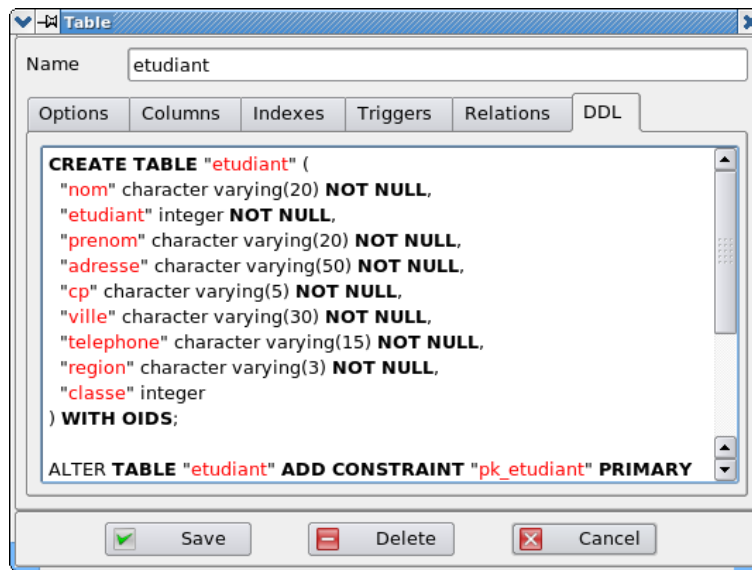
The third folder "Indices" presents a list of all the indices on the table, and allows through buttons appear, access to information on individual indices of the table itself. Double click on a line on the list behaves like button amendment, opening the dialog for editing properties of the index.



The fourth folder "Triggers" presents a list of all the triggers linked to the table, allowing through buttons appear, access to information on individual objects. Double click on a line on the list behaves like an edit button, opening the dialog box for editing properties of the object. This mode is added to the standard features of editing trigger, which will be described below.



The fifth tab "Relations" presents a tree-structured list, which displays any reports of the table with the other items contained in the draft. The list is for informational purposes only, and has no function.



The last folder, exist in almost all the windows on object-oriented database, appear SQL commands on the building and the construction of instruction takes place in real time, and depends on the settings in the property contained in the current window.

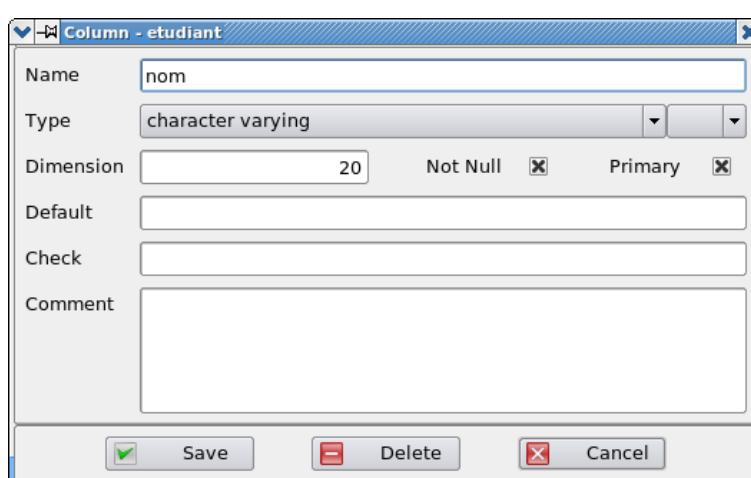
Table Fields

For technical specifications on the fields of a table, see the official documentation of *PostgreSQL*.

The window shown in the figures below, allows you to edit the properties of a field of a table.

As with all objects in a project, the area has a window where you must enter the name of the object, unique within the table. In addition to the name, the window contains all the properties of the field.

Changing a table field is accessible only by editing window table.



The combo "Type" contains a list of all data types allowed for the release of *PostgreSQL* selected. Tooltip If the option is enabled by placing the mouse combo will see a brief description of the type of selected field. Turning on the check "Primary key" results in the activation of the check "not null", and creating a name for the primary key in the editing of the table.

The "size" contains the size of the field. The content is treated as text, whose value must be composed of the entire party and the decimal part, separated by a comma; this is not necessary if the decimal value is zero.

Note: Changes made on the camps are not annulled if you cancel the changes made in the editing of the table.

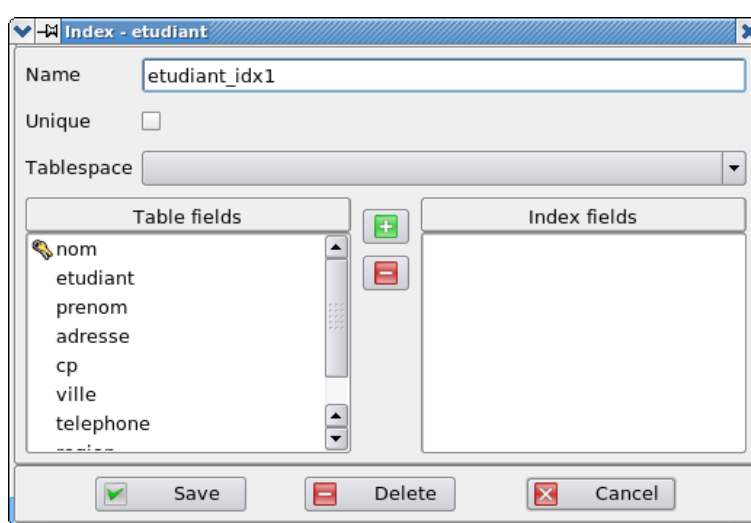
Table Index

For technical specifications on the indices of a table, see the official documentation of *PostgreSQL*.

The window shown in the figures below, lets you change the properties of an index of a table.

As with all objects in a project, the area has a window where you must enter the name of the object, unique within the table. In addition to the name, the window contains all the properties of the field.

Changing a table index is accessible only by editing window table.



The combo "Tablespace" contains a list of all objects tablespace in the project. By selecting one, it is hooked to it is in any case impossible not to assign a specific tablespace, in this case, the index will be assigned to the tablespace default *PostgreSQL*.

For the composition of index fields, there are two lists: the list on the left contains a list of all the fields in the table, while the list on the right lists all fields included in the index. Through the two buttons positioned between the two lists you can add or delete a field in the index ("+" adds, "-" delete).

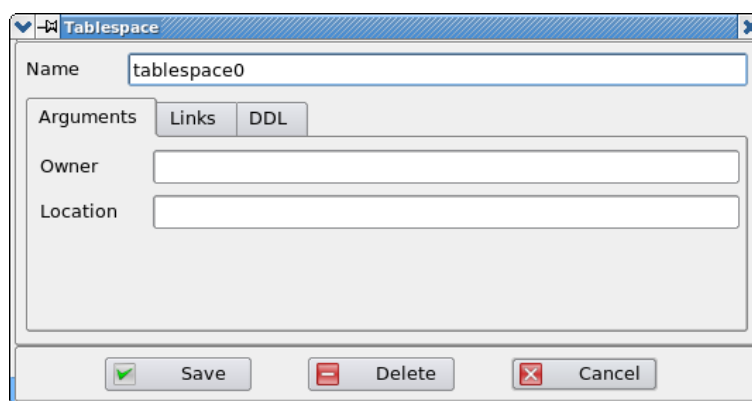
Note: Changes made on the indices are not annulled if you cancel the changes made in the editing of the table.

Tablespace

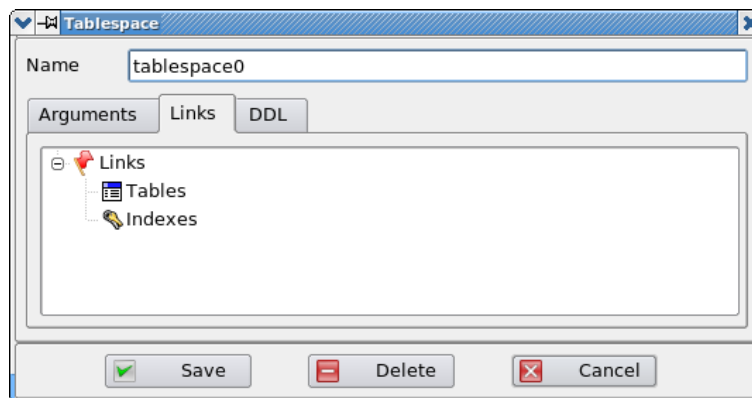
For technical specifications related tablespaces can be found in the official documentation of *PostgreSQL*.

The window shown in the figures below, lets you change the properties of an object tablespace.

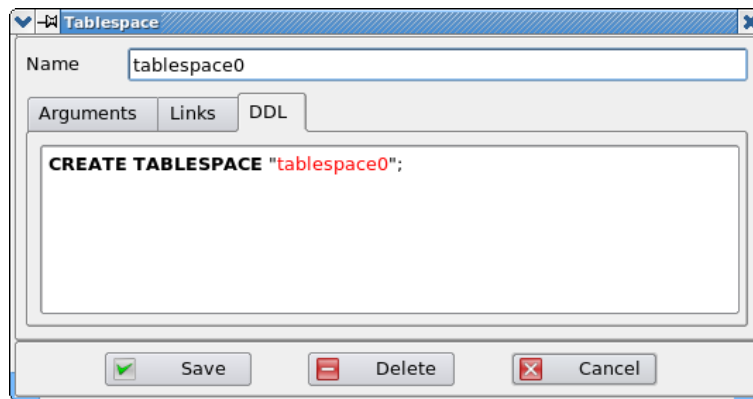
As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project. The window is made up of three folders, grouping in the different logical characteristics of the object.



In the first folder "Topics" are all properties of the object.



The second board "Links" presents a tree-structured list, which displays any reports of tablespace with the other items contained in the draft. The list is for informational purposes only, and has no function.



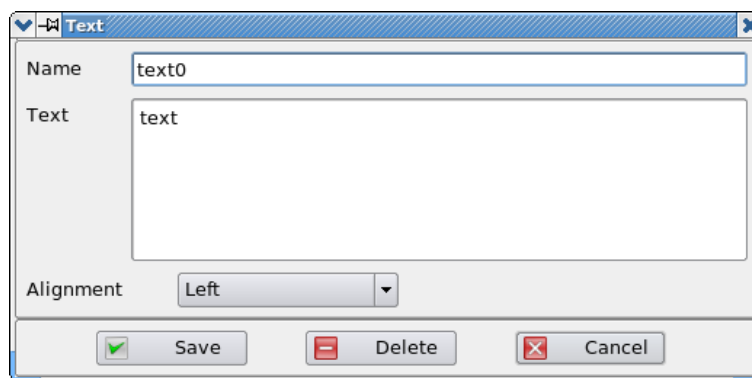
The last folder, exist in almost all the windows on object-oriented database, appear SQL commands on the building and the construction of instruction takes place in real time, and depends on the settings in the property contained in the current window.

Text

As for the objects of type Area and Stamp, the subject has Text purposes only within the graphic diagram of the project. Its use could be to describe an area of the diagram, or as a comment of the project shown in the graph.

The editing window of this type of object, as shown in the figures below, allows you to modify the properties.

As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project.



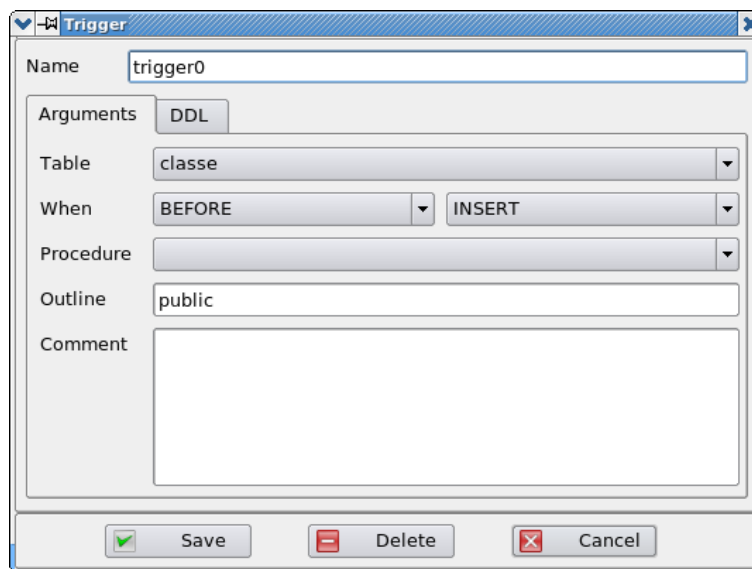
As previously described, the subject Text is a simple descriptive text for the properties consist in the content of this text and, optionally, as it is aligned within its space graph. Unlike a subject area, the text can be composed of several lines, then the window is a field lines eligible to enter the text components; space needed for the construction of the graphics is determined by the size of these lines and his number, while alignment determines the visual.

Trigger

For technical specifications on the trigger, see the official documentation of *PostgreSQL*.

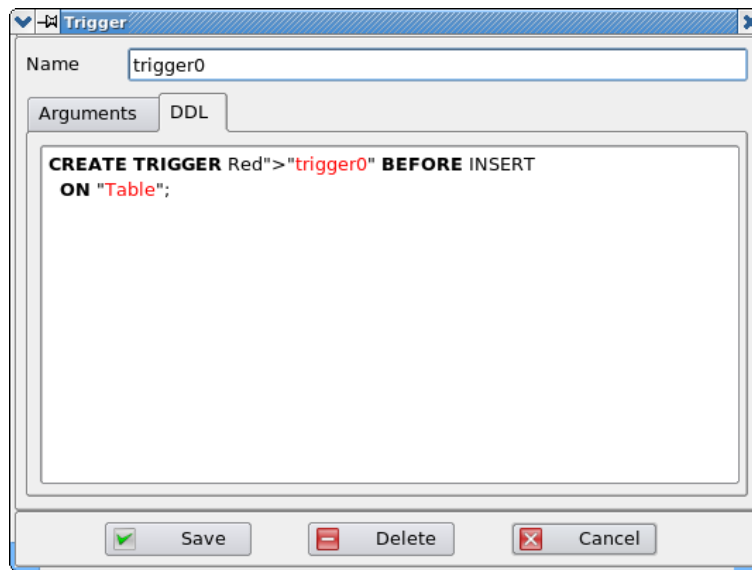
The window shown in the figures below, lets you change the properties of an object triggers.

As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project. The window is made up of two folders, grouping in the different logical characteristics of the object.



The folder "Topics" contains all the properties, including its references to other objects in the project. The combo "Table" contains a list of all types of objects contained in the draft table, while the combo "Procedures" contains a list of all objects procedure.

Note: It is not possible to create an object in the project trigger if there are no tables or procedures.



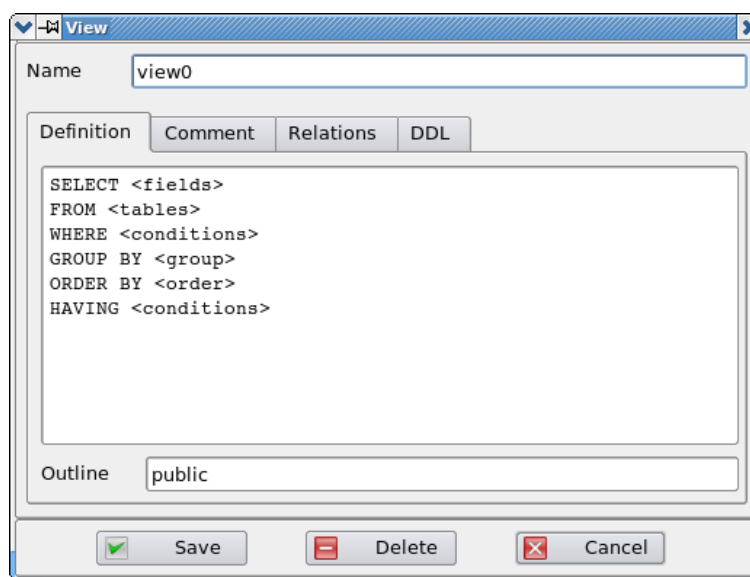
The last folder, exist in almost all the windows on object-oriented database, appear SQL commands on the building and the construction of instruction takes place in real time, and depends on the settings in the property contained in the current window.

View

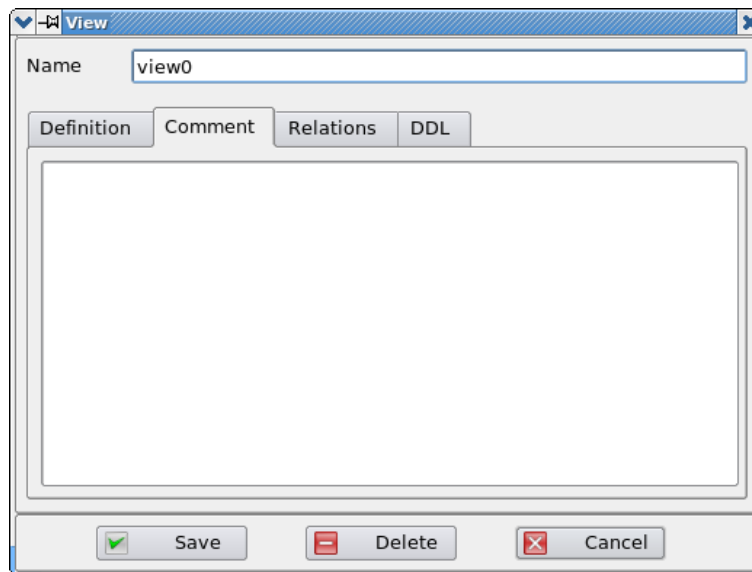
For technical specifications regarding views can be found in the official documentation of *PostgreSQL*.

The window shown in the figures below, lets you change the properties of an object view.

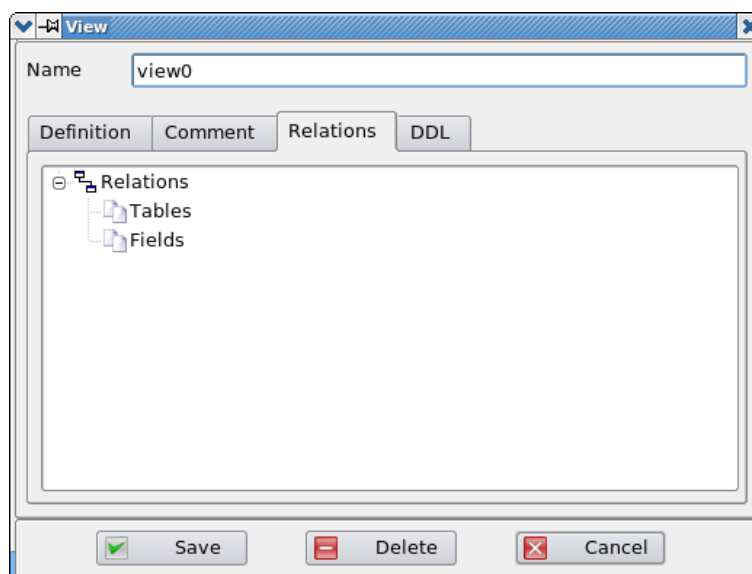
As with all objects in a project, the area has a window where you must enter the name of the object, unique within the same project. The window is made up of four folders, grouping in the different logical characteristics of the object.



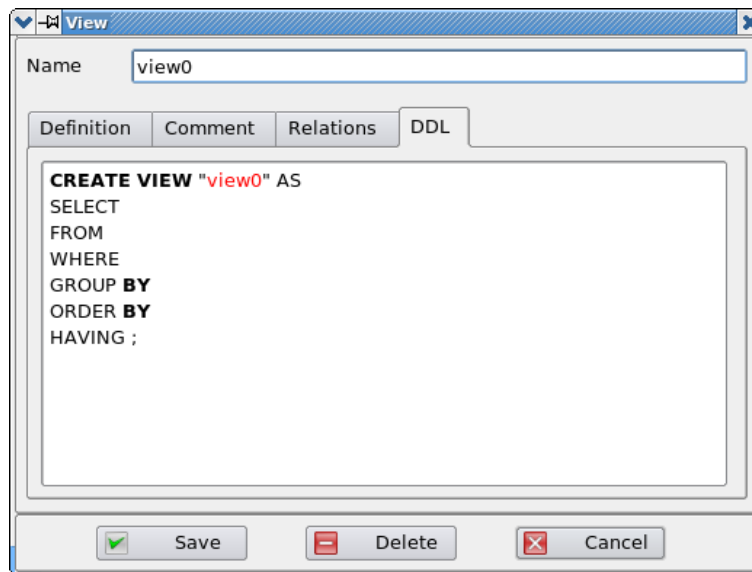
In Folder "Definition" you can enter the query on the SQL view. Basically, the procedure does not provide any control over the content of the query, but it is extracted the list of fields, in order to view them as detailed in the diagram of the project. An error in setting SELECT, also determines in an interpretation of this list, and thus a wrong representation in the graph. For new items are added automatically track the basis of SQL in order to facilitate the construction.



As with many other database objects, it is possible to define a description of view, which will be used in constructing the *PostgreSQL* database.



The third tab "Relations" presents a tree-structured list, which displays Any reports to the tables and views in the project, and detected by definition sight. The list is for informational purposes only, and has no function.

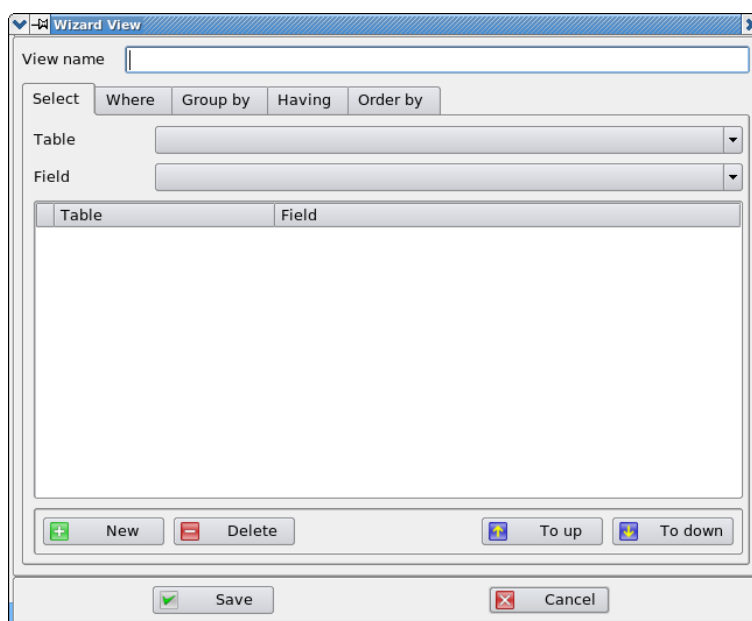


The last folder, exist in almost all the windows on object-oriented database, appear SQL commands on the building and the construction of instruction takes place in real time, and depends on the settings in the property contained in the current window.

View Creation Wizard

By creating views, *pgDesigner* provides a wizard that allows you to create an object view to a process driven. From the main menu or from the panel of instruments on the left of this window, you can access and start the wizard that, through a window containing a number of folders, to allow the integration of data necessary for creation the view.

As with all objects in a project, the area has a window where you must enter the name of the object that will be created, which must be unique within the same project. The window is made up of five folders, corresponding to commands that normally make a SQL SELECT.



In the first folder "Select" You can insert fields of tables and other views, which will form part of the query sight. As you can see from the figure, the folder contains four buttons with which you can add or delete fields, or move to a compliance order. At the center is a list that lists all fields belonging to the query, including references to the tables or views from which they were derived, the inclusion of a new field is carried out by selecting the table (or view) of origin in combo "Table" at the top tab, and one of the fields contained in it, acting on the second combo. Each change made in the first combo, the second will be upgraded automatically loading in its list all fields relating to the new selection.

The screenshot shows the 'Wizard View' dialog box with the 'Where' tab selected. The 'View name' field is empty. Below the tabs, there are two rows for defining conditions. The first row has 'Table' and 'Field/Value' dropdowns, an '=' operator, and a second 'Table' dropdown. The second row has 'Table' and 'Field/Value' dropdowns, an 'AND' operator, and a '+' button. Below these is a table with columns: Table, Field, Operator, Table, Field/Value, and a '+' button. The table is currently empty. At the bottom, there are buttons for '+ New', '- Delete', 'To up', 'To down', 'Save', and 'Cancel'.

View name

Select Where Group by Having Order by

Table

Field/Value

Operator =

Table

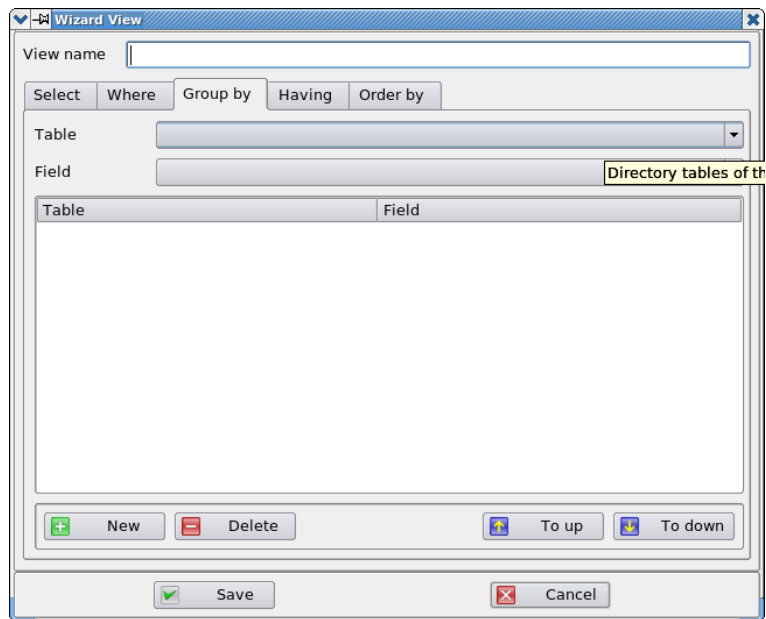
Field/Value

AND

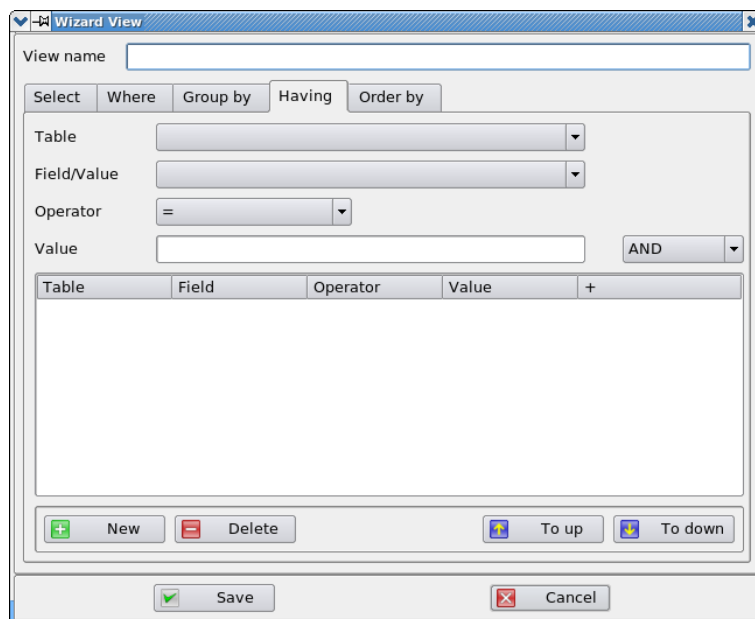
Table	Field	Operator	Table	Field/Value	+
-------	-------	----------	-------	-------------	---

+ New - Delete To up To down Save Cancel

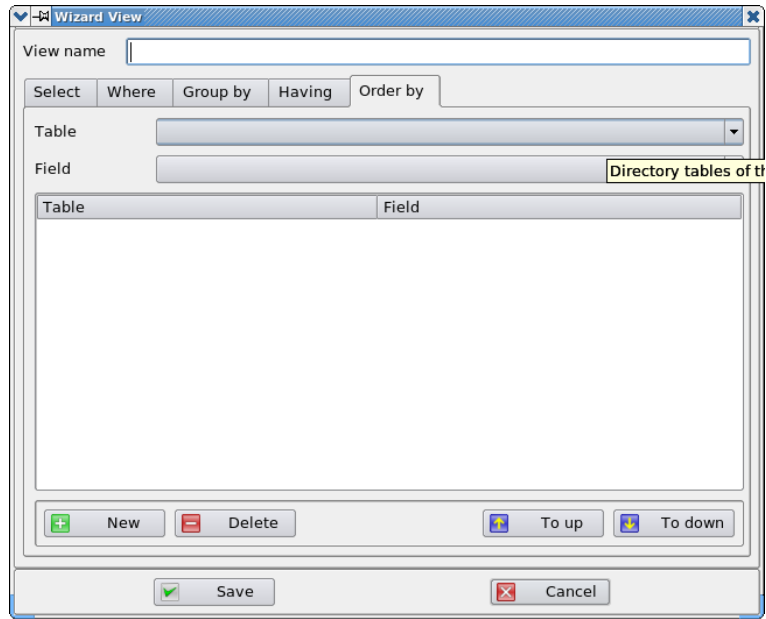
In the second tab "Where" is possible to define the conditions of the query filter. Based on the selected references in the previous card, you can set the conditions by which the view perform the extraction of data. Through the present controls tab you can also insert calculated fields (eg SUM), acting appropriately on controls. The listed below tab presents a list of conditions included in the query, and as for the previous card, you can insert or delete rows or order conditions in a different way.



The third tab "Group by" lets you enter the grouping necessary if, for example, the inclusion of a calculated field running the sum of value of a given field (SUM). As in previous, even here there are the usual buttons: insert a new grouping, erase an existing order, the sequence different sort of elements in the list.



The fourth tab "Having" can be used for placing conditions, other than those managed by WHERE. For a detailed description of the standard SQL syntax, please refer to other documentation.



The last card "by Order" allows you to define the order that will be provided from queries on the data before making them available. Like the other cards, this presents the usual buttons management list.

Other functions

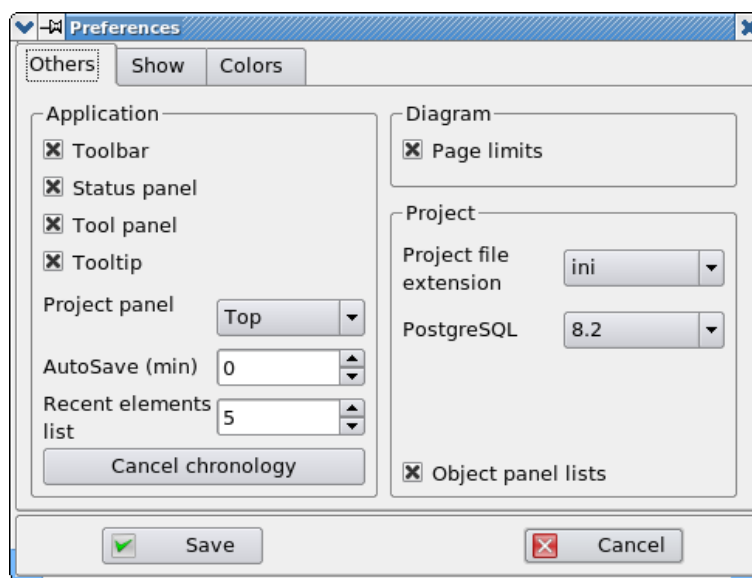
In the paragraphs contained in this chapter describes some features *pgDesigner* not tied to a project, but are used or are to assist in the management of the application.

Configuring pgDesigner

The settings general *pgDesigner* are enclosed in a separate window, accessible via the main menu. Through this window, you can change both the look and feel of open and projects, both default parameters for new projects.

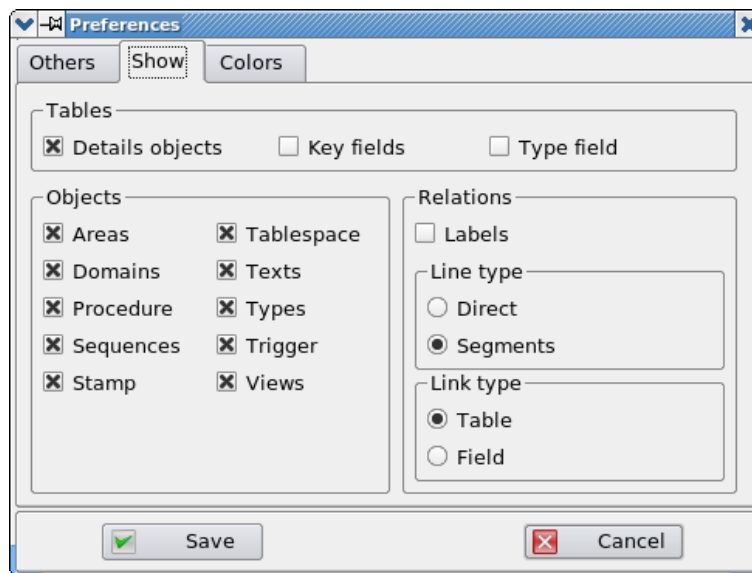
The window consists of a series of folders containing the general settings and current *pgDesigner*, observed by the general configuration file, which will be discussed in later chapters.

As you can see in the figure below, in the Miscellaneous folder, there are a variety of settings, defining both the overall look and feel of *pgDesigner*, the conditioning of some features of the background, or to configure the default values for new projects.



- Check the "Toolbar" lets you hide the panel under the menu;
- Check "Status Panel" allows you to hide the lower panel in the main window;
- Check the "Panel tools" lets you hide the panel of instruments on the left side of the main window;
- Check "Tooltip" enables the display of descriptions that appear when you mouse over an object (eg a button);
- The combo "Panel projects" determines the position of tabs panel projects;
- "AutoSave" is used to be a procedure in the background, which performs automatic saving projects changed. The value contained in this field determines the period of time between a rescue and the next, the value is configurable with steps of 5 minutes, a value of zero disables functionality;
- The value contained in the "elements in the list" determines the maximum number of historical items stored in the configuration file, and used to populate the menu Recent under the main menu, which is used to direct the opening of a project on this disc;

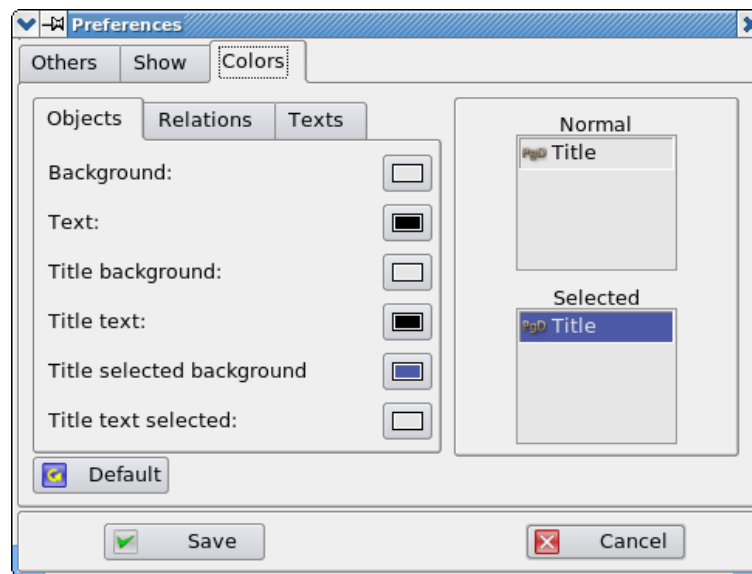
- Check the "Limits page" feature enables the design of dividing lines, or limits, page within the diagram, in order to see how the graph could be divided into being printed. These limits are designed in the form of dotted lines of light gray, in order not to interfere with visual objects contained in the diagram; coordinates and size of the resulting areas are determined based on the current print settings, and the corresponding size and the orientation of the paper print.
- The "file extension project" defines the default extension for the rescue of new projects; this value is purely indicative and subject to constant change, being influenced by loading procedures and rescue projects;
- The combo "*PostgreSQL*" determines the default database driver, the setting is used as a default for new projects, but has to bear in mind that could affect the loading of projects from sql files, which do not provide precise identity database destination;
- Check the "Panel list items" affects visibility panel located to the right of the diagram of a project, the condition is not essential, since the size of this panel to complete discretion, even when using the application.



In the next folder, the second from the left, contains some settings that affect the visibility of objects inside of the projects. Specifically:

- For tables you can hide the details, or the list of fields contained in the same table, or decide whether or not to display the only key primary, or even simply view the type of each field; Not all combinations are possible, in fact if you choose not generally see the details of the fields in your table, you can not select the check only key fields. Check also affects the display of other items, for example for the relations we can see whether or not the names of objects (in the form of labels placed at the center of each line), or for procedures display or hide parameters entry and exit;
- Check the contents in the "Objects" allow you to hide certain types of object, with the exception of tables and relations in the absence of which the plot would be meaningless;
- In "Relations" there are some settings to the type of display you want to configure only for relations;

check "Labels" lets you hide the labels with the names of relations on the diagram, while "Type line" and "Type hook" you can determine how they are drawn lines relation. If the "Type line" is set to "direct", the line follows a direct line from the table of origin to the destination table, but if it is set to "Segment" line is presented as a series of horizontal and vertical segments perpendicular. If "type hook" to be imposed on "Table", the line will start from the bottom of the table of origin and ending on top of the table location, but if it is set to "Field", the line starts from the right side at the first field used in the report in the establishment of origin and ending on the left side at the first field used in the report of the destination table.

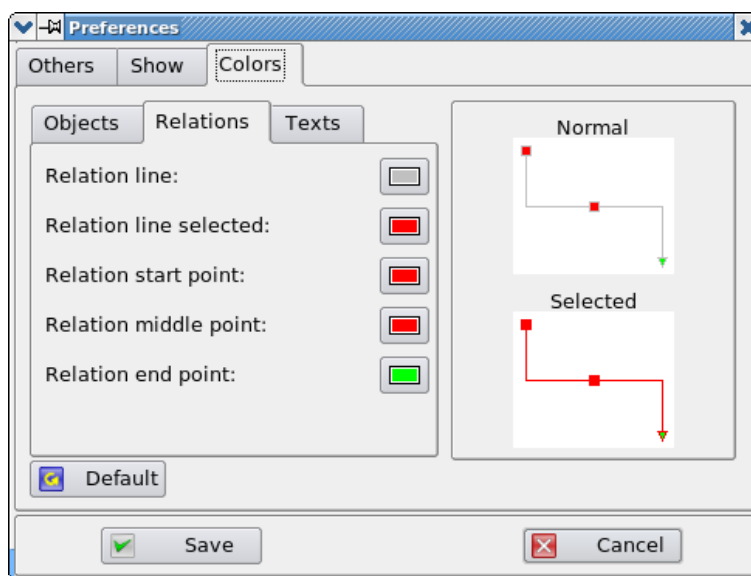


In the third folder, the file "Colors", there are three other folders in which you can define the colors of individual types of objects designed in the diagram. It should be borne in mind, and will be described later in later chapters, which *pgDesigner* inside defines three classes of objects, according to the approach and graphics from which it is put into the project generally objects (tables, etc.), relations, and texts; for the zones (an object purely chart), the color settings are included in the same customized and individually.

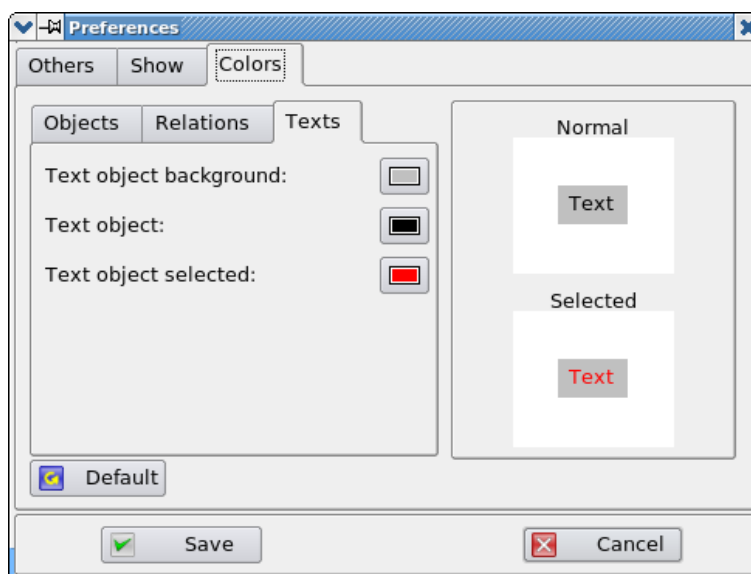
Each color folder contains within it a list of descriptions, references to many aspects graphics for the type of object that is gearing treating every reference to the right on a button, the color of which reflects the current setting of the property described. On the right of the window is a panel which, according to the type of object that is gearing setting, graphically shows an example of how this will be displayed in the diagram. To change a color simply press the button corresponding to the property to be amended, so as to open a dialog box, where you can define the new color, including through the establishment of RGB values. The amendment, once confirmed, will be shown in the example. At the end of configuration, and save the new settings will be presented with a confirmation message, which will be possible to determine whether the new approach will be applied to all projects open on only active project, or simply stored for the next new projects. It is considered that if the changes are confirmed and applied to the projects, they write above all customizations made on individual items, and will be saved with the project, otherwise any customization performed on individual objects, loading a project, they take precedence over the default settings.

In Folder "Objects" are the settings of color for all items, except relations, areas, and text. Apart from these particular groups of objects, everything else is graphically as a window: a title (the name of the object) in a

rectangle similar to the title of a window on the desktop, and a light gray, contains details of the object (in the case of a table, the list of fields ...).



In the second folder "Colors", there are the color settings for the objects of type relation. Objects of type relation occur in the form of lines, whose continuity is conditioned by the type of relationship (mandatory, not mandatory, etc.) Between tables. The point of attack with the source table (Table father) is defined with a small square of full color, while the point of attack on the destination table (Table daughter) is designed as a bit of arrow always full of color; in exactly half this line is also a small rectangle full color, and if I check "Label" is enabled, next to it the name of the report. The color settings for both the points attache that the same line.



The last folder "Texts" can define colors of the objects of type text. As you can see from the figure, these objects are simple text charts contained in a small area of size-dependent text. In subsequent chapters this type of object will be described in more detail.

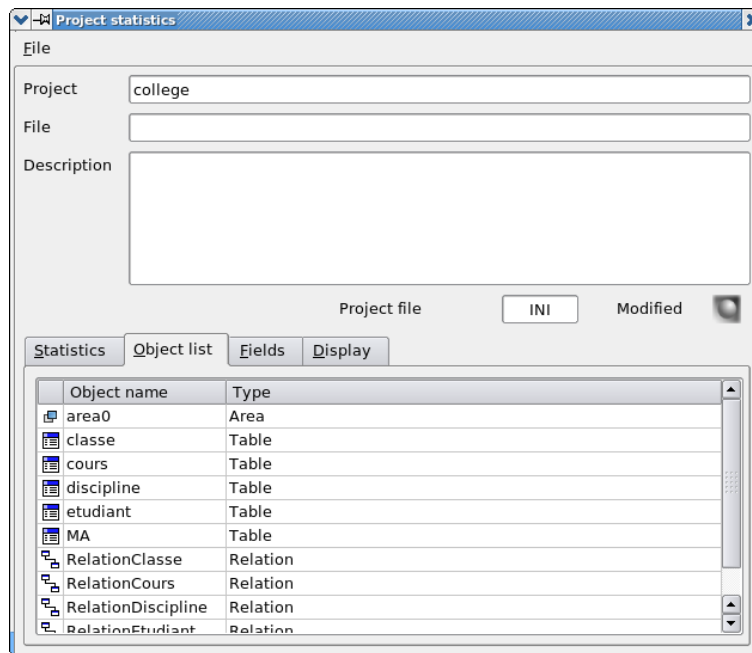
The window Statistics

The statistics window contains a wealth of information tied to the current status of a project. As you can see in the figure below, in the top of the window displays some of the information that is present in the parameters of the project, namely: the name of the project, the name of the file containing the project, the project description, extending used for file and an icon (or LEDs) to indicate the state to amend current (red = amended).

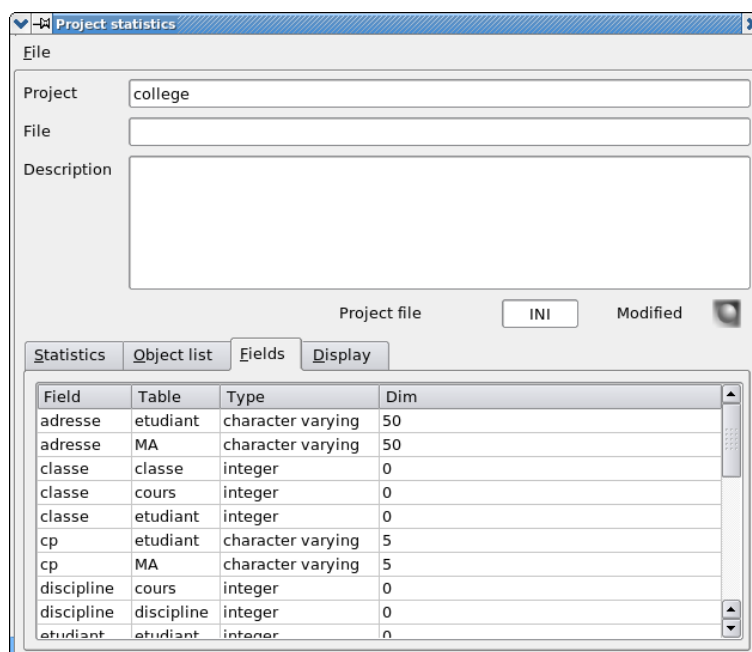
In below are a series of folders containing some statistical information: Statistics List objects, fields, display.

Count objects (11 total objetos)	
Domains	0
Procedures/Funcion	0
Relations	4
Sequences	0
Tables	5
Table fields	25
Table indexes	0
Tablespace	0
Texts	0
Types	0
Trigger	0
Views	0
Graphic areas	1
Stamp	1

In Folder "Statistics" are the values corresponding to the number of elements, divided by type, contained in the draft.

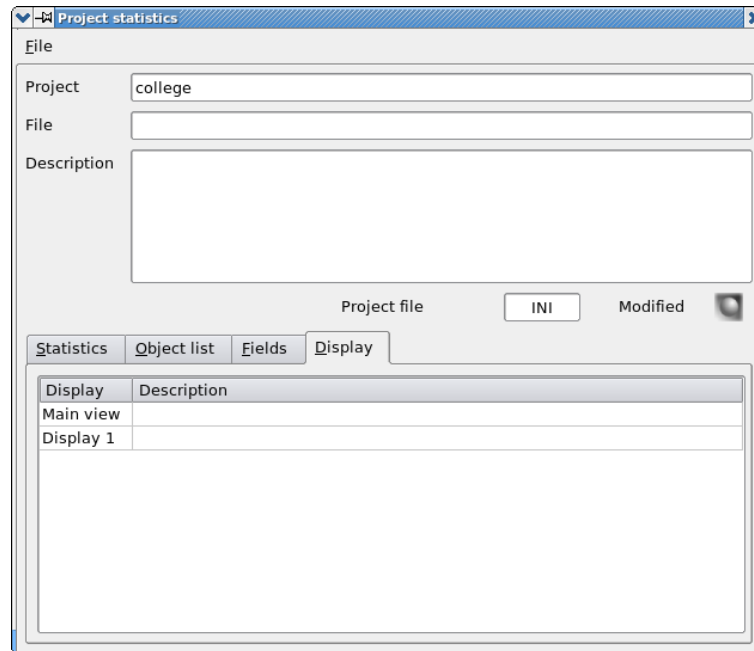


The folder "List items" list contains the name and type of all objects in the project. The list also has a function, the double click of the mouse on a line listed makes active on the object corresponding diagram and opens the window for editing properties of the object itself.



The folder "Fields" presents a list with the names of all the fields in all tables in the project, in addition to the name of the field are also displayed references to the name of the table they belong, the type and size of the

field. As for the list contained in the folder prior to the feature list also active, double-click the mouse on a line listed makes active on the table corresponding diagram and opens the window for editing properties of the object itself.

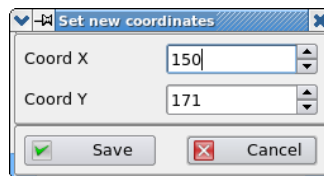


The last folder "Display" contains a list of display content in the project. As with the previous two folders list has a function, the double click of the mouse makes the display corresponding active, displaying the current diagram.

The statistics window also has a menu with which you can save to file or printer to send lists of items described.

Changing location coordinates of an object

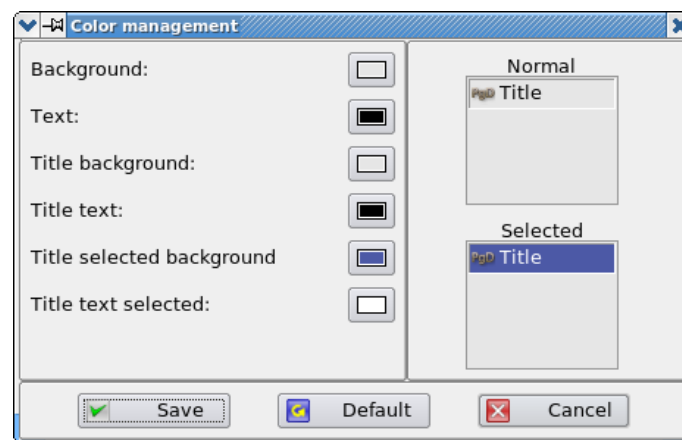
To change the position of an object in the plot, even taking into account the active display, there are several options: moving through mouse (pressing the left button and dragging on the desired coordinates), or by setting coordinates manually . In the second case, the function can be activated by opening the menu popup by first selecting the object with the mouse, or double click on the panel are state at the camp where they are usually displayed at the current coordinates of the object selected, all cases must first proceed to the selection of the object. The motion requires coordinated by the use of a dialog box, on which there are two fields to X and Y coordinates of the upper left corner compared to the diagram, using these two fields, you can enter the new coordinates, bearing in mind the size of the current and diagrams, so as not to place the object outside the graphics.



Customizing color of an object

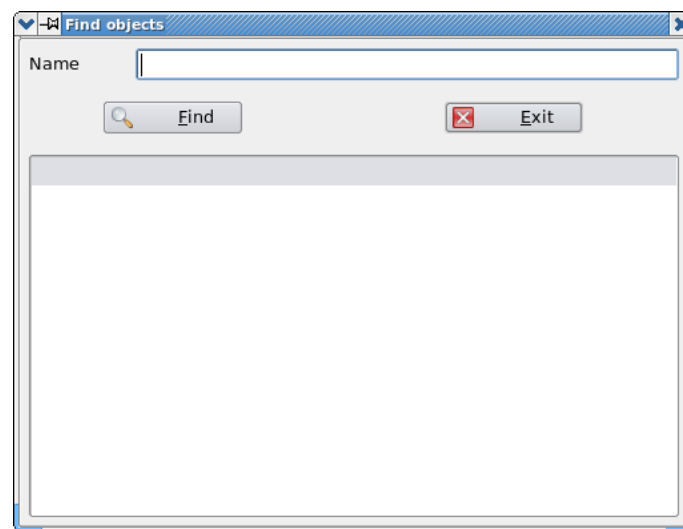
As for the general settings, which you can define globally colors of the objects, it is also possible to customize these settings for a single object. The function is activated by selecting the object and opening the popup menu, depending on the type of object, the window is presented that can take several aspects but in any case is very much like that of the general options. Even in this window are two distinct areas, left panel contains a list of settings for the object color and buttons for editing, while on the right is present examples of the panel, which is displayed in the preview issue graph object with the new color settings.

As described above, custom settings on a single subject will be undone if the setting is changed general.



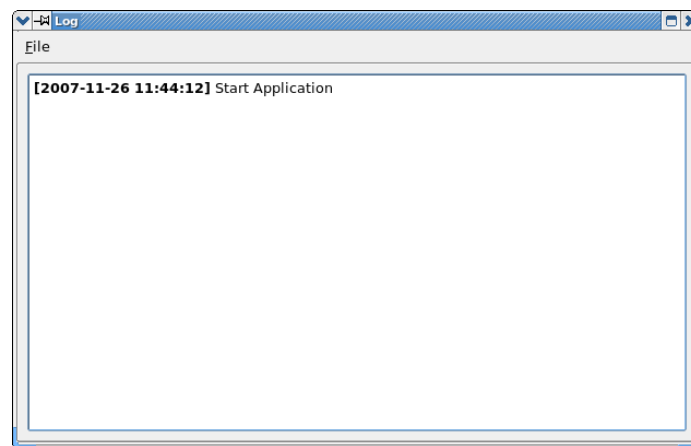
Finding objects within a project

In *pgDesigner* was implemented a search function, which allows users to find specific items within a project. The window is activated from the main menu and displays a simple field in which you can type your name, or part thereof) to search, a search for partial or on partial names, you can use the character '*'. After the search, if found correspondence, the list will display a list of objects found and where there are (for example a field, the name of the table that contains). Double click of a mouse on the object select line in the diagram and opens its editing window.



Viewing of the log

During its operation, *pgDesigner* stores the current status of a special log file. You can see the contents of this file by opening the window from the main menu. The contents of the window can be exported to another file or sent directly to the printer via menu items in the same window.



The log file is plain text, and designed to contain references to the date and time of the event and a description of the event.

Note: Every restart *pgDesigner* the file is clear, therefore, the information prior to the last launch will be lost in case of anomalies could be useful to have available the contents of the log, so you should save the file in order to transmit it in attached to a possible reporting.

The log file is contained in the directory `.pgdesigner/log`, in HOME directory of user.

Serious errors

If there is a serious anomaly in the use of *pgDesigner*, the window is presented in the following figure shows. This window contains the error code found, and a brief description of the location where it occurred. If bugs, the information will be useful to identify and correct the error, so it will be useful to refer to these data when reporting the bug.

Note: The closure of the application window will be finished anyway, and the unsaved changes will be lost.



General Configuration

By default *pgDesigner* in his first start, creates a hidden directory in HOME. The directory is named **.pgdesigner**. Inside are stored all files needed to *pgDesigner*, so be very careful not to delete this directory and its contents.

The configuration file

For all the default settings, configurations and the base of application, *pgDesigner* leans to a file called configuration files. This file is usually created at the first execution of *pgDesigner* and updated automatically every variation of the basic settings.

The first time you run *pgDesigner* in directory described in this chapter, you create the file **pgdesigner.conf**, which contains all the basic settings, configurations and user customization for the application. Any change in configuration *pgDesigner*, is updated and stored in this file.

The file is in a format similar to the classics .ini file with the settings grouped in special blocks, and every variable, along with their value, as defined in a single line in the following format: variable = value.

In the following paragraphs will be given a detailed description of the contents of this file and the various settings and values.

Structure of configuration files

The configuration file, as already mentioned, contains within it the default *pgDesigner*, which are updated from time to time to vary the properties within the application.

The structure of the file is made up of a series of clusters, each of which contains a number of properties and their values.

The table below describes in detail all groups and elements contained in the file, including the description of the possible values that ogni single property can take.

Group	Element	Type	Default	Description
CONNECTION				Group settings connection to the database. It contains the parameters of the connection.
	Host	String	<blank>	Name or address PostgreSQL server.
	Login	String	<blank>	User login.
	Name	String	<blank>	Name database.
	Port	String	5432	Port Door connection to the database..
IDE				Group basic settings of the program.
	AutoSave	Numeric	0	Determines the time between automatic saving projects and the next. A value equal to 0 disables the function.
	DimTreeEntity	Numeric	180	The width in pixels of panel containing the list of objects of the project.
	LastPath	String	<path utente>	Last directory saving / loading.
	ProjectPanelPosition	Numeric	Top	Determine the location of the labels panel projects.
	RelationAttach	Numeric	0	Tipo di aggancio delle relazioni alle tabelle: 0 = collegamento alle tabelle 1 = collegamento ai campi
	RelationType	Numeric	0	Type line designed for the relation: 0 = segmented 1 = Direct
	ShowArea	Logic	1	View objects of type Area (0 = false, 1 = true).
	ShowDetail	Logic	1	Show details objects (0 = false, 1 = true).
	ShowDomain	Logic	1	View objects of type Domain (0 = false, 1 = true).
	ShowFieldTypes	Logic	1	Show type fields of the tables (0 = false, 1 = true).
	ShowKeys	Logic	1	Show the primary keys for objects table (0 = false, 1 = true).
	ShowPages	Logic	0	View dividing lines in the pages of diagram of the project (0 = false, 1 = true).
	ShowProcedure	Logic	1	View objects of type procedure / function (0 = false, 1 = true).
	ShowRelationName	Logic	1	Show labels with the names of objects Relation (0 = false, 1 = true).

Group	Element	Type	Default	Description
	ShowSequence	Logic	1	View objects of type Sequence (0 = false, 1 = true).
	ShowSplash	Logic	1	Show splash screen at boot program (0 = false, 1 = true).
	ShowStamp	Logic	1	View objects of type Stamp (0 = false, 1 = true).
	ShowStatusPanel	Logic	1	Display panel status (0 = false, 1 = true).
	ShowTablespace	Logic	1	View objects of type Tablespace (0 = false, 1 = true).
	ShowText	Logic	1	View objects of type Text (0 = false, 1 = true).
	ShowToolBar	Logic	1	Display the menu bar (0 = false, 1 = true).
	ShowToolBox	Logic	1	Display panel tools (0 = false, 1 = true).
	ShowTooltip	Logic	1	Show tooltip aid (0 = false, 1 = true).
	ShowTreeEntity	Logic	1	Show list objects of the project (0 = false, 1 = true).
	ShowTrigger	Logic	1	View objects of type Trigger (0 = false, 1 = true).
	ShowType	Logic	1	View objects of type Type (0 = false, 1 = true).
	ShowView	Logic	1	View objects of type View (0 = false, 1 = true).
PRINTER				Group printer settings.
	Orientation	Numeric	0	Orientation printer: 0 = Portrait 1 = Landscape
	Size	String	A4	Size sheet printer.
PROJECT				Group basic settings of the new projects.
	Driver	String	8.0	Last PostgreSQL driver used, eg. "8.2".
	Extension	String	ini	Extension default file project.
	LastProject	String	<blank>	Last open project.
OPTIONS				Group options import / export.
	ExportCreateIndex	Logic	1	Enable export indices table (0 = false, 1 = true).
	ExportCreatePrimaryKey	Logic	1	Enables the creation of primary keys (0 = false, 1 = true).
	ExportDomain	Logic	1	Activate the export of the objects domain (0 = false, 1 = true).
	ExportIncludeComment	Logic	1	Activate the export of comments (0 = false, 1 = true).
	ExportIncludeDrop	Logic	1	Enable export command drop objects (0 = false, 1 = true).
	ExportProcedure	Logic	1	Activate the export of the objects procedure (0 = false, 1 = true).
	ExportRelation	Logic	1	Activate the export of the objects relation (0 = false, 1 = true).
	ExportSequence	Logic	1	Activate the export of the objects sequence (0 = false, 1 = true).
	ExportTable	Logic	1	Activate the export of the objects table (0 = false, 1 = true).
	ExportTablespace	Logic	1	Activate the export of the objects tablespace (0 = false, 1 = true).
	ExportTrigger	Logic	1	Activate the export of the objects trigger (0 = false, 1 = true).
	ExportType	Logic	1	Activate the export of the objects type (0 = false, 1 = true).
	ExportView	Logic	1	Activate the export of the objects view (0 = false, 1 = true).
	ImportCreateIndex	Logic	1	Activate the importation of indices table (0 = false, 1 = true).

Group	Element	Type	Default	Description
	ImportCreatePrimaryKey	Logic	1	Activate the import of primary keys (0 = false, 1 = true).
	ImportDomain	Logic	1	Activate the importation of objects domain (0 = false, 1 = true).
	ImportIncludeComment	Logic	1	Activate the import of comments (0 = false, 1 = true).
	ImportIncludeDrop	Logic	1	Enable import Command drop objects (0 = false, 1 = true).
	ImportProcedure	Logic	1	Activate the importation of objects procedure (0 = false, 1 = true).
	ImportRelation	Logic	1	Activate the importation of objects relation (0 = false, 1 = true).
	ImportSequence	Logic	1	Activate the importation of objects sequence (0 = false, 1 = true).
	ImportTable	Logic	1	Activate the importation of objects table (0 = false, 1 = true).
	ImportTablespace	Logic	1	Activate the importation of objects tablespace (0 = false, 1 = true).
	ImportTrigger	Logic	1	Activate the importation of objects trigger (0 = false, 1 = true).
	ImportType	Logic	1	Activate the importation of objects type (0 = false, 1 = true).
	ImportView	Logic	1	Activate the importation of objects view (0 = false, 1 = true).
WINDOW				Group settings of the main window.
	Maximized	Logic	0	Indicates that the main window must be expanded to the size of the screen (0 = false, 1 = true).
	Size	String	0,0,800,600	Size of the main window in the screen coordinates, in the following format: left, top, width, height.
RECENT				Group historic elements.
	MaxRecent	Numeric	5	Maximum number of elements for the historic projects.
	Project<n>	String	<blank>	One of the last five open projects, complete with path.
OBJECT				Group basic settings for the objects.
	BackColor	String	230,230,230	Background color objects (size: r, g, b).
	TextColor	String	0,0,0	Color text objects (size: r, g, b).
	TitleBackColor	String	230,230,230	The background color of title of the objects (size: r, g, b).
	TitleBackColorSelected	String	76,89,166	The background color of title when selected objects (size: r, g, b).
	TitleForeColor	String	0,0,0	Color of Title objects (size: r, g, b).
	TitleForeColorSelected	String	230,230,230	Color of Title objects when selected (size: r, g, b).
	RelationLine	String	192,192,192	Color line relation (size: r, g, b).
	RelationLineSelected	String	255,0,0	Color line when selected relation (size: r, g, b).
	RelationStartPoint	String	255,0,0	Color initial point of relation (size: r, g, b).
	RelationMiddlePoint	String	255,0,0	Color central point of the report (size: r, g, b).

Group	Element	Type	Default	Description
	RelationEndPoint	String	0,255,0	Color endpoint of the relation (size: r, g, b).
	WTextForeColor	String	0,0,0	Color text object "text" (size: r, g, b).
	WTextBackColor	String	192,192,192	Background Color object "text" (size: r, g, b).
	WTextForeColorSelected	String	255,0,0	Color text object "text" when selected (size: r, g, b).

The File Project

The project file is a text file, structured so as to include all the information and objects relating to the project itself, so that we can rebuild the entire plot within *pgDesigner*.

According to the configuration program, you can load and save files in special projects, with two different formats: INI and XML. As described in previous chapters, the contents of two files is the same, but with different structures which are stored data; because of its structure is possible that the phases of reading and rescue deadlines have very different, even if equal data. In any case, the choice of using either type are at the discretion of the user, bearing in mind that you can always revert the trend, as all both possibilities available.

As already mentioned, the project file containing all the basic settings for a single project, objects and views, and the positioning of graphical objects within them.

In the following pages describes the technical specifications relating to the structure of these files, depending on their type.

Structure File Project

INI Format

The INI format, as in the configuration file of the program is a classic format with the appropriate settings grouped into blocks, and each variable (or characteristic of the object) with its value, each described in a single line, in the format : variable = value.

Except for some of the features, dedicated to the graphic objects of the project relate to specific properties *PostgreSQL*, which refer to the official documentation for the investigation of the case.

Group	Element	Type	Default	Description
PgDesigner				Group features of the project file.
	Version	String	<blank>	Identifies version of the file. The official version is the release of pgDesigner; in latest release has been revised methodology and diverse classes numbering between the application and project files. Where possible has been maintained a certain compatibility with earlier versions, but this does not exclude the probability an old project can not be read with the new release of pgDesigner.
PROJECT				Group characteristics of the project.
	Author	String	<blank>	Author's name. It contains the name of the project.
	Description	String	<blank>	Project Description. It can contain a brief description of the project.
	Display	Numeric	0	Number of current display. Indicates the number of active display. Immediately after loading the project, the application sets this display as current display, showing only the diagram objects contained in this display.
	Driver	String	8.2	Number Identification version of PostgreSQL. The value determines the characteristics of the project, and determines the type of database that you want to create.
	Name	String	<blank>	Project Name (required).
	Orientation	Numeric	0	Orientation of the press: 0 = Portrait 1 = Landscape The setting is used for the default configuration for prints, but can be changed within the application.
	Page	String	A4	Page Format Printing. The setting is used for the default configuration for prints, but can be changed within the application.
	Revision	String	<blank>	Given the last change in format: dd/mm/yyyy hh:mm:ss. The value does not drive the project, but is only used as an indicator and is updated every time you make a change to the project.
OBJECTS				Contains the number of objects present in the file, separated by type.

Group	Element	Type	Default	Description
	Area	Numeric	0	Counter. This indicates the total number of objects of type area in the file.
	Domain	Numeric	0	Counter. This indicates the total number of domain objects of type in the file.
	Text	Numeric	0	Counter. This indicates the total number of objects of type text in the file.
	Procedure	Numeric	0	Counter. This indicates the total number of objects of type procedure / function in the file.
	Relation	Numeric	0	Counter. This indicates the total number of objects in relation type in the file.
	Sequence	Numeric	0	Counter. This indicates the total number of objects of type sequence in the file.
	Table	Numeric	0	Counter. This indicates the total number of objects of type table in the file.
	Tablespace	Numeric	0	Counter. This indicates the total number of objects of type tablespace in the file.
	Trigger	Numeric	0	Counter. This indicates the total number of objects trigger type in the file.
	Type	Numeric	0	Counter. This indicates the total number of objects of type "type" in the file.
	View	Numeric	0	Counter. This indicates the total number of objects of type view, contained in the file.
AREA_<n>				Contains features of a single type of subject area. Each object is identified by the label "AREA_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of the area.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	W	Numeric	0	Width of the object.
	H	Numeric	0	Height of the object.
	Display	Numeric	0	Number display to which the object is associated.
	Color	String	0,0,0	Background color of the object (size: r, g, b).
	Description	String	<blank>	Text content in the object.
TEXT_<n>	Bold	Logic	0	Printing bold (0 = normal, 1 = bold).
				Has the characteristics of a single object type text. Each object is identified by the label "TEXT_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of the text.

Group	Element	Type	Default	Description
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Alignment	String	L	Text alignment: L = Left C = centered R = Right
	Value	String	<blank>	Contents of the text.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorFore	String	0,0,0	Color text (size: r, g, b).
	ColorForeSelected	String	0,0,0	Color text when selected (size: r, g, b).
TABLESPACE _<n>				Has the characteristics of a single object type tablespace. Each object is identified by the label "TABLESPACE_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of tablespace.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Location	String	<blank>	The directory used for the tablespace.
	Owner	String	<blank>	Username owner of the tablespace.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
TYPE				Contains features of a single type of object "type". Each object is identified by the label "TYPE_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of the object type.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Input	String	<blank>	Name function that converts the data from an external type to a type of procedure.
	Output	String	<blank>	Name function that converts data from one type to a type internal external.
	Receive	String	<blank>	Name function that converts the data from an external binary type to a type of procedure.

Group	Element	Type	Default	Description
	Send	String	<blank>	Name function that converts data from one type to an internal binary type outside.
	Analyze	String	<blank>	Name function that performs the statistical analysis for the data type.
	InternalLength	Numeric	0	Constant number that specifies the length in bytes of internal representation of the new type.
	PassedByValue	Logic	0	Indicates that the values of this type of data are passed by value or by reference. Values allowed: 0 or 1.
	Alignment	String	<blank>	Alignment of memory required for the data type. If specified, it must be char, int2, int4, or double.
	Storage	String	<blank>	Strategy memory for the data type. If specified, it must be plain, external, extended, or main.
	DefaultValue	String	<blank>	The default value for the type of data.
	Element	String	<blank>	The type is created in an array; specifies the type of the array elements.
	Delimiter	String	<blank>	Character used delimiter between the values built into arrays of this type.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
SEQUENCE_ <n>				Has the characteristics of a single object type sequence. Each object is identified by the label "SEQUENCE_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of sequence.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Cache	Numeric	0	Optional clause specifying how many sequential numbers must be preallocated and stored for fast access.
	Increment	Numeric	0	Value of increase. The property is optional and specifies what value is added to the current value of the sequence for the creation of a new value.
	Initial	Numeric	0	The property is optional to activate the sequence starting from this value.
	Iterate	Logic	0	The option allows the restart sequence numbering to achieve maximum or minimum values. Values allowed: 0 or 1.
	MaxValue	Numeric	0	The clause is optional and determines the maximum value of the sequence.
	MinValue	Numeric	0	The clause is optional and determines the minimum value of the

Group	Element	Type	Default	Description
				sequence.
	Comment	String	<blank>	Description object.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
PROCEDURE _<n>				Has the characteristics of a single object type procedure / function. Each object is identified by the label "PROCEDURE_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of procedure / function.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Argument	String	<blank>	The name of an argument of the function.
	Argument2	String	<blank>	The type of the function's arguments, if any.
	Definition	String	<blank>	A string defining the content of the function, the internal structure is dependent on the language used.
	Language	String	<blank>	The name of the language used in the implementation of the function.
	Property1	String	<blank>	This attribute inform the system to save any multiple assessments function, treating them as a single. Values admitted: IMMUTABLE, unstable or VOLATILE.
	Property2	String	<blank>	Mode function call. Values admitted: "CALLED ON INPUT NULL" or "RETURNS NULL NULL ON INPUT".
	Property3	String	<blank>	Privileges of function call: Values admitted: "SECURITY INVOKER" or "SECURITY DEFINER".
	ReturnValue	String	<blank>	The type of the return value.
	ReturnValue2	String	<blank>	The type of the return value.
	Comment	String	<blank>	Description object.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).

Group	Element	Type	Default	Description
DOMAIN_<n>				Contains features of a single type of object domain. Each object is identified by the label "DOMAIN_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of the domain.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Check	String	<blank>	The clause specifies the integrity constraints or verify what domain must be satisfied.
	DefaultValue	String	<blank>	The clause specifies a default value for the columns of the type of domain.
	NullValue	String	<blank>	It allows you to enter null values.
	Type1	String	<blank>	The data type domain.
	Type2	String	<blank>	The data type domain.
	Type3	String	<blank>	The data type domain.
	Constraint	String	<blank>	The name of the optional constraint.
	Comment	String	<blank>	Description object.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
TABLE_<n>				Has the characteristics of a single object type table. Each object is identified by the label "TABLE_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of the table.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Comment	String	<blank>	Description object.
	Option	String	<blank>	The clause is optional and if specific lines of the table should have the object identifier (OID).
	PrimaryKey	String	<blank>	Specify whether the column or columns of a table must contain unique values (unduplicated).
	Tablespace	String	<blank>	The name of the tablespace where you create a new table.
	Field_<n>	String	<blank>	Each column is described in a single line, identified by a serial number that corresponds to the position in the table. Parameter column enclosed within double quotes, separated by a " ". The order parameter

Group	Element	Type	Default	Description
				column is as follows: 1) Field name 2) Description of the field 3) Value of default: assigns a default value for column 4) The size of the field 5) Flag enabling column containing null values 6) Primary key: specifies that the column must contain unique values (unduplicated) 7) The data type column 8) The data type column 9) Check: specifies an expression that returns a value logical lines when new or modified meet an input or update.
	Index_<n>	String	<blank>	Each index is defined in a single line, identified by a serial number. The parameters are in the enclosed within double quotes, separated by a " ". The order of the parameters is as follows: 1) Name index 2) Name of tablespace where the index is created 3) Unique: Specifies that the columns in the index must contain only unique values. 4) List the fields of the table, separated by commas.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
TRIGGER_<n>				Contains features of a single type of object triggers. Each object is identified by the label "TRIGGER_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of triggers.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	BeforeAfter	String	<blank>	Determines if the function is called before or after the event. It can take the following values: BEFORE or AFTER.
	InsUpdDel	String	<blank>	Specifies that the event should be triggered by the trigger. It can take the following values: INSERT, UPDATE or DELETE.
	Procedure	String	<blank>	The function name, declared without an argument and returns a value of type trigger, which is performed when the trigger is activated.
	Table	String	<blank>	The name of the table on which the trigger is activated.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).

Group	Element	Type	Default	Description
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
VIEW_<n>				Has the characteristics of a single object type view. Each object is identified by the label "VIEW_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of view.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Definition	String	<blank>	Definition SQL.
	Comment	String	<blank>	Description object.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
RELATION_<n>				Contains features of a single type of object relation. Each object is identified by the label "RELATION_", followed by a serial number. This number is set automatically by operation of rescuing the project.
	Name	String	<blank>	Name of the object.
	X1	Numeric	0	Coordinated X of the table than the father.
	Y1	Numeric	0	Coordinated Y of the table than his father.
	X2	Numeric	0	Coordinated X of the table than the daughter.
	Y2	Numeric	0	Coordinated Y of the table than the daughter.
	Table1	String	<blank>	Table Name father.
	Table2	String	<blank>	Name table daughter.
	OnUpdate	String	<blank>	The clause specifies the action to take when a line of the table is referenced amended.
	OnDelete	String	<blank>	The clause specifies the action to take when a line of the table is referenced canceled.
	Relation_<n>	String	<blank>	Any link between the two camps is defined in a single line, identified by a serial number that corresponds to the position in the same relation. The two fields are enclosed in double quotes, separated by a comma. The order is as follows:

Group	Element	Type	Default	Description
				1) Field table father 2) Field table daughter
	ColorLine	String	0,0,0	Color line (size: r, g, b).
	ColorLineSelected	String	0,0,0	Color line when selected (size: r, g, b).
	ColorStartPoint	String	0,0,0	Color initial point (size: r, g, b).
	ColorMiddlePoint	String	0,0,0	Color central point (size: r, g, b).
	ColorEndPoint	String	0,0,0	Color terminal point (size: r, g, b).
DISPLAYS				It contains all the features of the display (or views) of the project. The group is defined once, and all group and display objects contained in each of them.
	Display	Numeric	0	Total display defined in the project.
	Display[d]	String	<blank>	Each display is defined in a single line, identified by a serial number "d" enclosed in square brackets (as an array). The parameters of the display are enclosed in double quotes, separated by commas. The order of elements is as follows: 1) Identification Display 2) Name Display 3) Description display
	Display[d][n]	String	<blank>	Every object in the display (identified by progressive "d"), is defined in a single line, identified by a serial number "n". The parameters are enclosed in double quotes, separated by commas. The order of elements is as follows: 1) Name of the object in the display 2) Coordinated X inside of the display 3) Coordinated Y inside of the display

XML Format

The XML format, as in the format ini, is composed by elements of the project, the views and characteristics of the project itself. The settings of each object are defined in the attributes of each item. To simplify management, the structure is organized so as to reduce as far as possible the proliferate labels of the main elements, and placing all the features of the project and object attributes of each item. In this way an attempt was made to reduce the size of files, optimizing the functions of writing and reading, and avoiding as much as possible the problems related to over consumption of resources and system crashes that occurred in previous versions of *pgDesigner*.

All the elements are enclosed in a single element: *pgDesigner*.

Except for some of the features, dedicated to the graphic objects of the project relate to specific properties *PostgreSQL*, which refer to the official documentation for the investigation of the case.

Element	Attribute	Type	Default	Description
PgDesigner				Group features of the project file.
	Version	String	<blank>	Identifies version of the file. The official version is the release of pgDesigner; in latest release has been revised methodology and diversified the cassi numbering between the application and project files. Where possible has been maintained a certain compatibility with earlier versions, but this does not exclude the probability an old project can not be read with the new release of pgDesigner.
PROJECT				Group characteristics of the project.
	Name	String	<blank>	Project Name (required).
	PageFormat	String	<blank>	Page Format Printing. The setting is used for the default configuration for prints, but can be modified in the project.
	PageOrientation	String	A4	Orientation of the press: 0 = Portrait 1 = Landscape The setting is used for the default configuration for prints, but can be modified in the project.
	Revision	Numeric	0	Given the last change in format: dd/mm/yyyy hh:mm:ss. The value does not drive the project, but is only used as an indicator.
	Description	String	<blank>	Project Description. It can contain a brief description of the project.
	Driver	String	8.2	Number Identification version of PostgreSQL. The value determines the characteristics of the project, and determines the type of database that you want to create.
	Author	String	<blank>	Author's name. It contains the name of the project.
	Display	Numeric	0	Number of current display. Indicates the number of active display. Immediately after loading the project, the application sets this display as current display, showing only the diagram objects contained in this display.

Element	Attribute	Type	Default	Description
AREA				Has the characteristics of a single object, type area.
	Name	String	<blank>	Name of the area.
	Display	Numeric	0	Number display to which the object is associated.
	Color	String	0,0,0	Background color of the object (size: r, g, b).
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	W	Numeric	0	Width of the object.
	H	Numeric	0	Height of the object.
	Description	String	<blank>	Text of the object.
	Bold	Logic	0	Printing bold (0 = normal, 1 = bold).
TEXT				Has the characteristics of a single object type text.
	Name	String	<blank>	Name of the text.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Alignment	String	L	Text alignment: L = Left C = centered R = Right
	Value	String	<blank>	Contents of the text.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorFore	String	0,0,0	Color text (size: r, g, b).
	ColorForeSelected	String	0,0,0	Color text when selected (size: r, g, b).
TABLESPACE				Has the characteristics of a single object type tablespace.
	Name	String	<blank>	Name of tablespace.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Location	String	<blank>	The directory used for the tablespace.
	Owner	String	<blank>	Username owner of the tablespace.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
SEQUENCE				Has the characteristics of a single object type sequence.

Element	Attribute	Type	Default	Description
	Name	String	<blank>	Name of sequence.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Cache	Numeric	0	Optional clause specifying how many sequential numbers should be allocated and stored for fast access.
	Increment	Numeric	0	Value of increase. The property is optional and specifies what value is added to the current value of the sequence for the creation of a new value.
	Initial	Numeric	0	The property is optional to activate the sequence starting from this value.
	Iterate	Logic	0	The option allows the restart sequence numbering to achieve maximum or minimum values. Values allowed: 0 or 1.
	MaxValue	Numeric	0	The clause is optional and determines the maximum value of the sequence.
	MinValue	Numeric	0	The clause is optional and determines the minimum value of the sequence.
	Comment	String	<blank>	Description object.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
PROCEDURE				Has the characteristics of a single object type procedure / function.
	Name	String	<blank>	Name of procedure / function.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Argument	String	<blank>	The name of an argument of the function.
	Argument2	String	<blank>	The type of the function's arguments, if any.
	Definition	String	<blank>	One tightens that it defines the content of the function; the inner structure is employee from the used language.
	Language	String	<blank>	The name of the language used in the construction of the function.
	Property1	String	<blank>	This attribute informs the system to save eventual multiple appraisals of the function, dealing them like one single. Admitted values: IMMUTABLE, STABLE o VOLATILE.
	Property2	String	<blank>	Modality of call of the function. Admitted values: "CALLED ON NULL INPUT" or "RETURNS NULL ON NULL INPUT".
	Property3	String	<blank>	It privileges of call of the function: Admitted values: "SECURITY INVOKER" or "SECURITY DEFINER".

Element	Attribute	Type	Default	Description
	ReturnValue	String	<blank>	The type of the return value.
	ReturnValue2	String	<blank>	The type of the return value.
	Comment	String	<blank>	Description object.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
DOMAIN				Contains features of a single type of object domain.
	Name	String	<blank>	Name of the domain.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Check	String	<blank>	The clause specifies the integrity constraints or value as head of the domain must be satisfied.
	DefaultValue	String	<blank>	The clause specifies a default value for the columns of the type of domain.
	NullValue	String	<blank>	It allows you to enter null values.
	Type1	String	<blank>	The data type domain.
	Type2	String	<blank>	The data type domain.
	Type3	String	<blank>	The data type domain.
	Constraint	String	<blank>	The name of the optional constraint.
	Comment	String	<blank>	Description object.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
TABLE				Has the characteristics of a single object type table.
	Name	String	<blank>	Name of the table.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Comment	String	<blank>	Description object.

Element	Attribute	Type	Default	Description
	Option	String	<blank>	The clause is optional and if specific lines of the table should have the object identifier (OID).
	PrimaryKey	String	<blank>	Specify whether the column or columns of a table must contain unique values (unduplicated).
	Tablespace	String	<blank>	The name of the tablespace where you create a new table.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
TABLEFIELD				Definition of a field single table.
	Name	String	<blank>	Name field.
	Table	String	<blank>	Name of the table.
	Check	String	<blank>	Specifies an expression that returns a value logical lines when new or modified meet an input or update.
	Comment	String	<blank>	Description of the field.
	DefaultValue	String	<blank>	Assign a default value for the column.
	Length	String	<blank>	Size of the field.
	NullCheck	Logic	0	Flag enabling column containing null values.
	PrimaryKey	String	<blank>	Specifies that the column must contain unique values (unduplicated).
	Type1	String	<blank>	The data type column.
	Type2	String	<blank>	The data type column.
INDEX				Definition of a single index table.
	Name	String	<blank>	Name index.
	Table	String	<blank>	Name of the table.
	Tablespace	String	<blank>	Name of tablespace where the index is created.
	Unique	Logic	0	Specifies that the columns in the index should only contain unique values.
INDEXFIELD				Definition of a field index.
	Name	String	<blank>	Field Name index.
	Table	String	<blank>	Name of the table.
	Index	String	<blank>	Name index.
TRIGGER				Contains features of a single type of object triggers.
	Name	String	<blank>	Name of triggers.
	X	Numeric	0	Coordinated X of the object.

Element	Attribute	Type	Default	Description
	Y	Numeric	0	Coordinated Y of the object.
	BeforeAfter	String	<blank>	Determines if the function is called before or after the event. It can take the following values: BEFORE or AFTER.
	InsUpdDel	String	<blank>	Specifies that the event should be triggered by the trigger. It can take the following values: INSERT, UPDATE or DELETE.
	Procedure	String	<blank>	The function name, declared without an argument and returns a value of type trigger, which is performed when the trigger is activated.
	Table	String	<blank>	The name of the table on which the trigger is activated.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
TYPE				Contains features of a single type of object "type".
	Name	String	<blank>	Name of the object type.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Input	String	<blank>	Name function that converts the data from an external type to a type of procedure.
	Output	String	<blank>	Name function that converts data from one type to a type internal external.
	Receive	String	<blank>	Name function that converts the data from an external binary type to a type of procedure.
	Send	String	<blank>	Name function that converts data from one type to an internal binary type outside.
	Analyze	String	<blank>	Name function that performs the statistical analysis for the data type.
	InternalLength	Numeric	0	Constant number that specifies the length in bytes of internal representation of the new type.
	PassedByValue	Logic	0	Indicates that the values of this type of data are passed by value or by reference. Values allowed: 0 or 1.
	Alignment	String	<blank>	Alignment of memory required for the data type. If specified, it must be char, int2, int4, or double.
	Storage	String	<blank>	Strategy memory for the data type. If specified, it must be plain, external, extended, or main.
	DefaultValue	String	<blank>	The default value for the type of data.
	Element	String	<blank>	The type is created in an array; specifies the type of the array elements.
	Delimiter	String	<blank>	Character used delimiter between the values built into arrays of this

Element	Attribute	Type	Default	Description
				type.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
VIEW				Has the characteristics of a single object type view.
	Name	String	<blank>	Name of view.
	X	Numeric	0	Coordinated X of the object.
	Y	Numeric	0	Coordinated Y of the object.
	Definition	String	<blank>	Definition SQL.
	Comment	String	<blank>	Description object.
	ColorBack	String	0,0,0	Background Color (size: r, g, b).
	ColorText	String	0,0,0	Color text (size: r, g, b).
	ColorBackTitle	String	0,0,0	Background Color of Title (size: r, g, b).
	ColorBackTitleSelected	String	0,0,0	Background Color of Title when selected (size: r, g, b).
	ColorForeTitle	String	0,0,0	Color title (size: r, g, b).
	ColorForeTitleSelected	String	0,0,0	Color title when selected (size: r, g, b).
RELATION				contains features of a single type of object relation.
	Name	String	<blank>	Name of the object.
	X1	Numeric	0	Coordinated X of the table than the father.
	Y1	Numeric	0	Coordinated Y of the table than the father.
	X2	Numeric	0	Coordinated X of the table than the daughter.
	Y2	Numeric	0	Coordinated Y of the table than the daughter.
	Table1	String	<blank>	Table Name father.
	Table2	String	<blank>	Name table daughter.
	OnUpdate	String	<blank>	The clause specifies the action to take when a line of the table is referenced amended.
	OnDelete	String	<blank>	The clause specifies the action to take when a line of the table is referenced canceled.
	ColorLine	String	0,0,0	Color line (size: r, g, b).
	ColorLineSelected	String	0,0,0	Color line when selected (size: r, g, b).
	ColorStartPoint	String	0,0,0	Color initial point (size: r, g, b).

Element	Attribute	Type	Default	Description
	ColorMiddlePoint	String	0,0,0	Color central point (size: r, g, b).
	ColorEndPoint	String	0,0,0	Color terminal point (size: r, g, b).
RELATIONFIELD				Defining fields of the tables in relation. Each line determines the fields of a table with his father of the table.
	Relation	String	<blank>	Name of relation.
	Field1	String	<blank>	Field Name table father.
	Field2	String	<blank>	Field Name table daughter.
DISPLAY				Has the characteristics of a display (or view) of the project.
	Name	String	<blank>	Name of the display.
	Description	String	<blank>	Description of the display.
DISPLAYOBJECT				It defines an object in the display.
	Display	String	<blank>	Name of the display.
	Name	String	<blank>	Name of the object in the display.
	X	Numeric	0	Coordinated X inside of the display.
	Y	Numeric	0	Coordinated Y inside of the display.

License

pgDesigner is distributed under the GNU GENERAL PUBLIC LICENSE (see original text on the next page).

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work

based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

- a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
- b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software

interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the

rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free

Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the

"copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) year name of author

Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type `show w'.

This is free software, and you are welcome to redistribute it under certain conditions; type `show c' for details.

The hypothetical commands `show w' and `show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `show w' and `show c'; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program `Gnomovision' (which makes

passes at compilers) written by James Hacker.

<signature of Ty Coon>, 1 April 1989

Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.