

# Introduction to PostgreSQL 8.2 on Windows

Published by the Open Source Software at Microsoft, February 2007

Special thanks to Chris Travers, Contributing Author to the Open Source Software Lab

Most current version will be maintained at: <http://port25.technet.com>



# Introduction to PostgreSQL 8.2 on Windows

## Why?

PostgreSQL is a very powerful relational database manager which is increasingly being used for large open source business applications. Since version 8, it has had a native Windows port maintained with the MinGW toolkit. Perhaps you want to install an application that depends on PostgreSQL, or perhaps you want to explore writing such applications. This guide will help get you started. The software can be downloaded from <http://www.postgresql.org/ftp/binary/> and is released under the BSD license with no advertising clause.

The software is distributed as a zip archive that contains two MSI packages. You will need to extract them to a directory of your choosing and then double click on the postgresql-8.2-int.msi file.

## Windows-specific Notes:

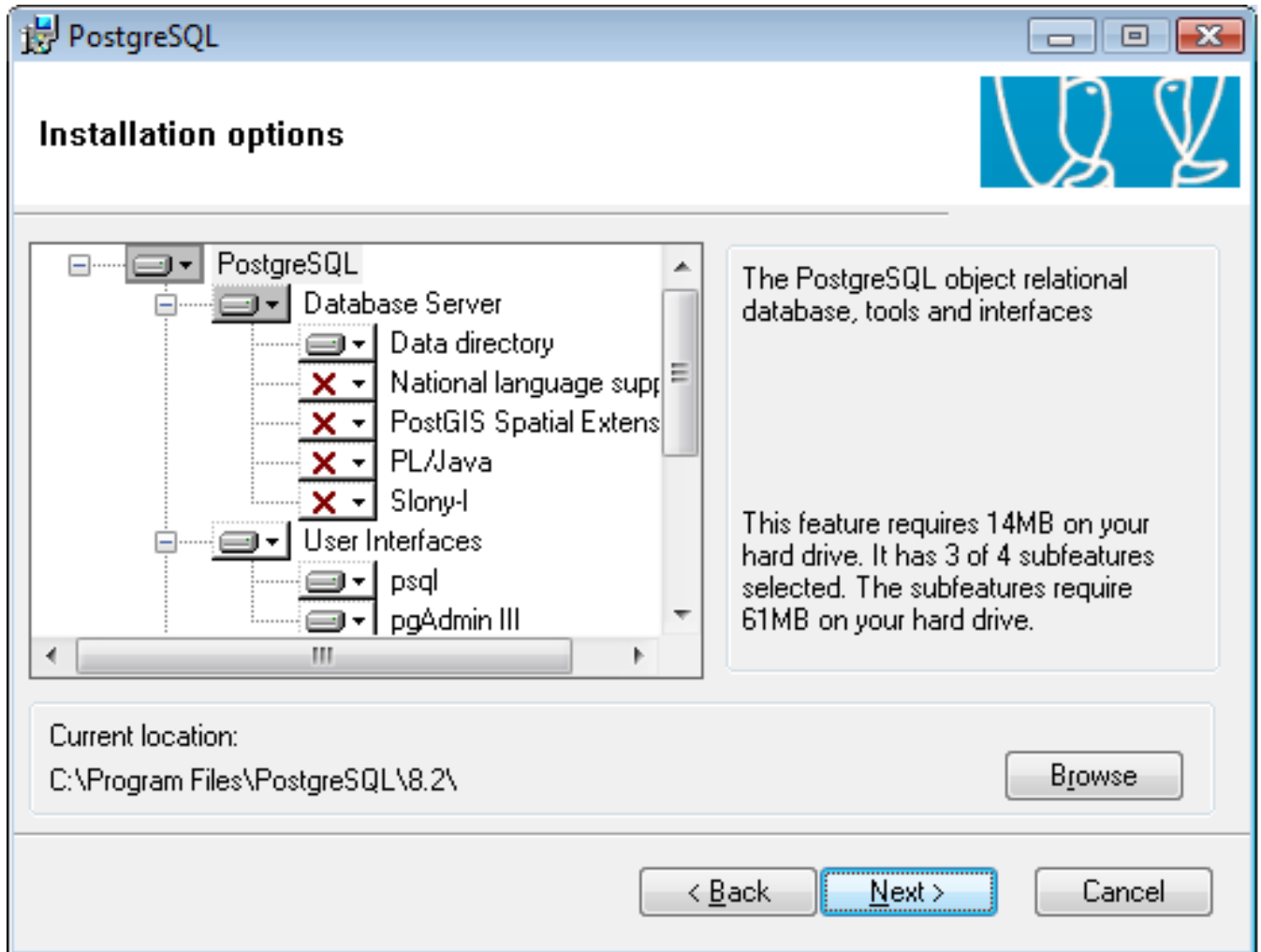
PostgreSQL uses a process-oriented architecture similar to that of Apache 1.3, where each request is handled by an independent process. The native Windows port has not used the Windows preferred thread-oriented architecture. For this reason, one should expect performance on Windows to be lower, especially where large numbers of small queries are executed.

Also not all add-in modules available from the community may be included out of the box. Creating more complex custom solutions with lots of off-the-shelf add-ons may be somewhat difficult on Windows at the moment.

## Installation

The installation of the software is fairly straight-forward. A few installation steps deserve some additional discussion, however.

A few screens into the installation one encounters the following screen regarding installation options:

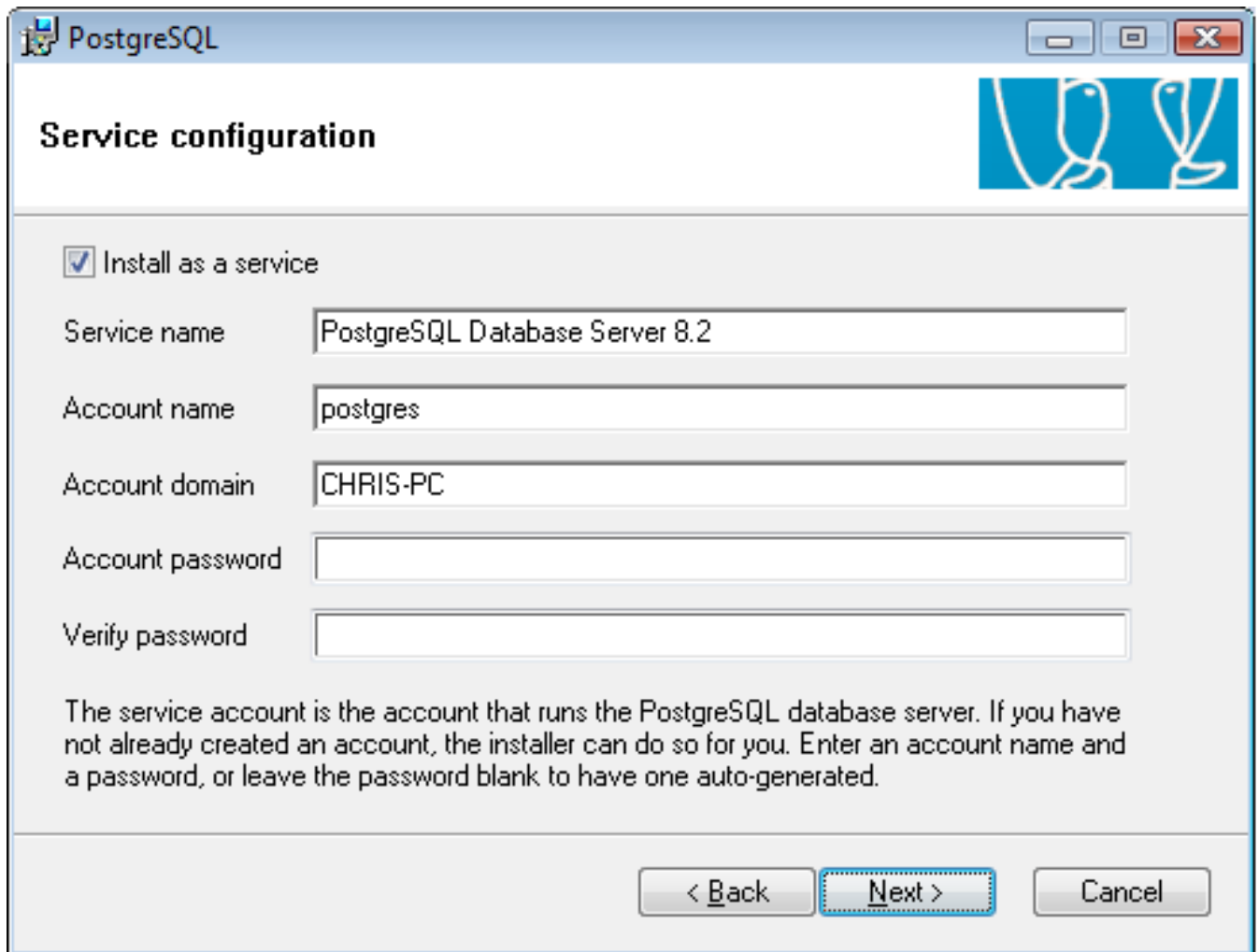


The components that can be included in the database server include:

- Data Directory sets up the data directory so it is ready to store information.
- National Language provides error and status messages localized for various languages.
- PostGIS Spatial Extensions provide spatial extensions used for geographic data.
- PL/Java is a handler that allows one to run stored procedures written in Java. Requires a JRE.
- Slony-I is a master-slave asynchronous replication system.

There are also administrative interfaces and development options. For the purposes of this introduction, I left all values at their defaults.

The next screen of interest is entitled "Service configuration:"



All account information refers to the system account that the database server will run as. Prior to version 8.2, PostgreSQL refused to run with administrative privileges as a security precaution designed to limit the damage caused by a security incident involving this software. Although running with elevated privileges is now supported, I recommend running the software as an ordinary user because this is better tested. If the account specified does not exist, the installer will offer to create it for you.

The next screen is entitled "Initialize database cluster:"

PostgreSQL

## Initialise database cluster

Initialize database cluster

Port number: 5432

Addresses:  Accept connections on all addresses, not just localhost

Locale: C

Encoding: SQL\_ASCII

Superuser name: postgres

Password:

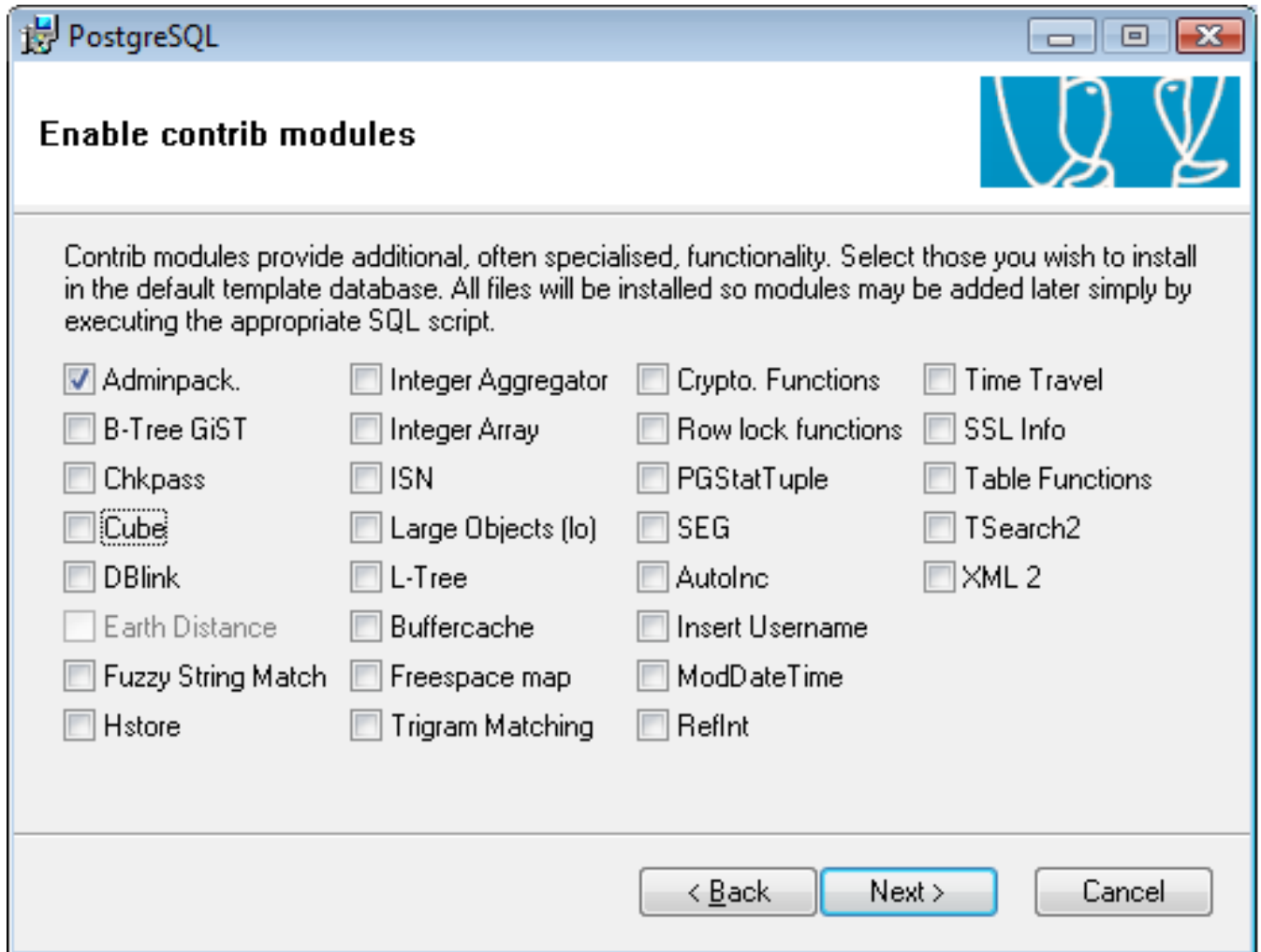
Password (again):

This is the internal database username, and not the service account. For security reasons, the password should NOT be the same as the service account.

< Back   Next >   Cancel

It is generally recommended that you change the encoding to UTF-8 unless you have strong reasons not to. This change is largely required if you want to store data from multiple character sets in a database cluster at any given time. However, it may cause issues if client applications are not Unicode-aware.

The final noteworthy screen is the "Contrib Modules" screen.



This contains a large number of extensions which can be installed at this time. These are auxiliary and is beyond the scope of this paper.

### Notes on Vista:

The install on Vista is similar to other Windows installs but to install on Vista, you must turn off User Account Control first.

## Configuration

PostgreSQL is originally a UNIX application and so the configuration system is entirely based on text files. These files can be accessed under the Start Menu -> All Programs -> PostgreSQL 8.2 -> Configuration files.

### Understanding the postgresql.conf file

The postgresql.conf file contains the main configuration parameters for the server. The file is well commented and most directives should be left at their defaults unless you know you need to

change them if the use is likely to be light to moderate. If heavier use is expected, one should adjust upwards the `max_connections` and `shared_buffers` up to suitable levels.

However, a few general settings are worth discussing.

## **FSYNC**

The `fsync` directive should generally be left on except in specific exotic circumstances. This option tells the server to flush the write-ahead log (WAL) to disk after every transaction. The idea is that once the log data is on disk, if the system crashes or loses power, the data in the tables can be recreated from the log entries. One should, however, bear in mind that many IDE drives do have caches on them which can allow for partially written WAL segments in the event of a crash or power outage. Disabling the write cache on the drive will depend on what version of Windows one is using but can generally be done from the disk drive properties section of the device manager.

## **Autovacuum**

The Windows installer package for PostgreSQL turns on a feature called "Autovacuum" which helps preserve performance as data is updated. This is necessary because PostgreSQL uses a lock-less concurrency system called MVCC or Multi-Version Concurrency Control. In this system, old rows remain in existence in case a transaction is rolled back. This allows faster rollbacks, but all data ever stored in the database (whether or not it is visible) until removed by a "vacuum" routine. This process sorts through a table, sets transaction id's on visible data into the indefinite past, and adjusts the free space map so that data that is no longer valid can be overwritten. The autovacuum option allows one to let PostgreSQL manage this process so that the administrator doesn't have to do it manually. One would generally recommend leaving it enabled.

## **Understanding the `pg_hba.conf` File**

PostgreSQL has an ability to use external sources of authentication including Kerberos, and LDAP. The authentication source is determined by the host-based authentication subsystem and is configured in this file. The file is well commented and the external authentication options are beyond the scope of this paper.

The internal authentication types are:

- "trust" allows the matching user to authenticate without further credentials being checked. This can be useful if the superuser password is lost, but it is not recommended for use in production.
- "reject" denies the request of the matching user without further checking any credentials.
- "password" requests the password in clear text. It is not recommended for production use except where client libraries do not support better authentication.
- "crypt" uses the `crypt()` function to encode the password. It is not recommended for production use because it is incompatible with the way passwords are encrypted on the server (and so requires plain text passwords to be stored there).

- "md5" is the preferred internal authentication method for PostgreSQL. It is also the default.

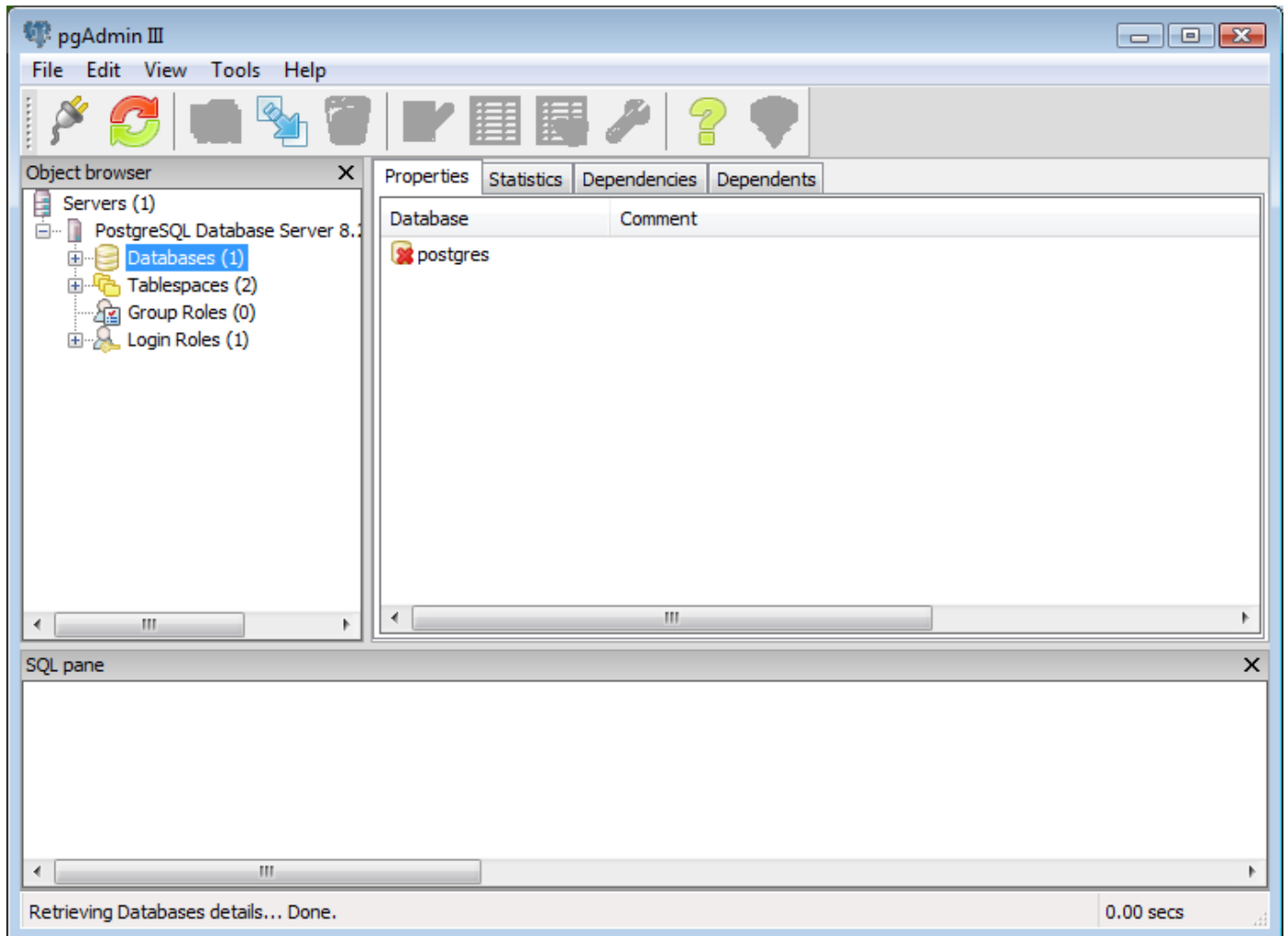
Each entry consists of a line beginning with "local," "host," "hostssl," or "hostnossl." This keyword describes the type of connection request. Local connections correspond to UNIX domain sockets and are not supported on Windows. The host-based options reference TCP/IP connections optionally with or without SSL. The first relevant entry found which matches the connection request determines which method is used to authenticate the user.

Finally, the default pg\_hba.conf only allows connections from localhost, even if the system is listening on an external address. If you want to make your database accessible to applications running on other systems, you must add host entries in this file with appropriate CIDR addresses.

## Connecting with PgAdmin III

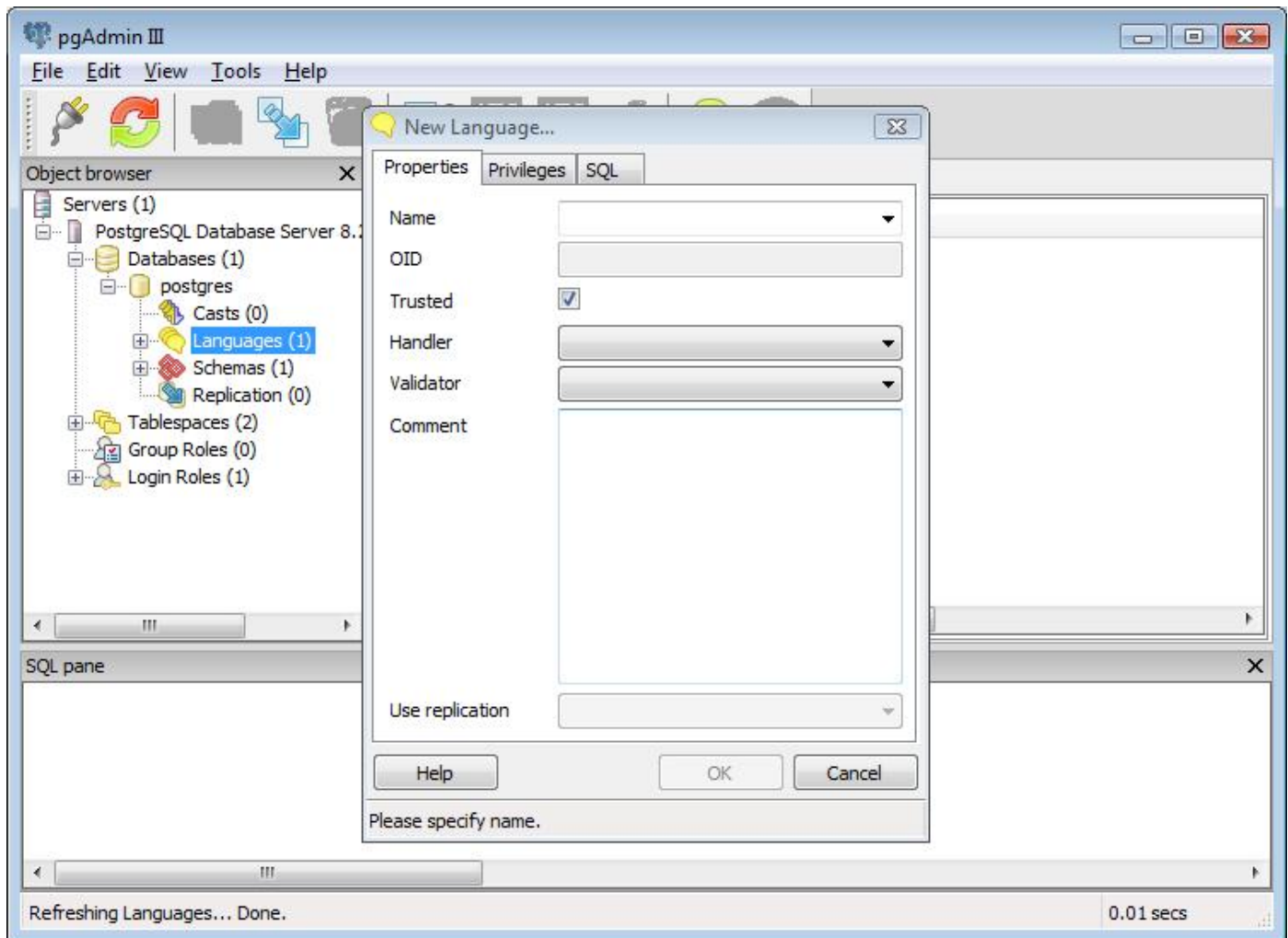
PGAdmin III is a nice graphical front-end for PostgreSQL database administration. While it is a separate project, it is bundled with the PostgreSQL Windows Installer package.

Once you start the application, you can double click on the server you wish to connect to. You will be prompted for a password if that is required by the server. Once you are connected, the screen will look similar to the following screen shot:



From here there are two ways to create a database. The first method is to right-click on "Databases" and select "New Database." The other way is to single click on "Databases." From there, you can select the "Create" option on the "Edit" menu.

The default PostgreSQL installation does install PL/Pgsql as a language for stored procedures (SQL and C are other built-in languages). If, however, you only want to install the language in some databases, you can forgo that option in the installation process and create the language in the database. To do this, first expand the database list by double clicking it. Then double click the desired database in order to expand it. Then right-click on "Languages" and select "New Language." The following pop-up screen will appear:



In the "Name" field, select "plpgsql." The rest of the fields will be grayed out. Click "OK."

Note that other procedural languages can be installed too but they generally require external dependencies.

## Conclusions

This paper provides an introduction to the use of PostgreSQL on Windows. While it does not cover many advanced topics relating to the administration and use of the software, this should be enough information to get the software up and running, as well as avoid some of the most common problems with deployment.

## About the Author

Chris Travers has over seven years experience working with PostgreSQL as his primary RDBMS of choice. He is the owner of Metatron Technology Consulting, a business devoted to helping customers use open source software and a core contributor LedgerSMB, an open source, web-based accounting package which relies on PostgreSQL for data storage and maintenance.

## Copyright

Information in this document, including URL and other Internet Web site references, is subject to change without notice and is provided for informational purposes only. The entire risk of the use or results from the use of this document remains with the user, and Microsoft Corporation makes no warranties, either express or implied. Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

© 2007 Microsoft Corporation. This work is licensed under the Microsoft Permissive License. The Microsoft Permissive License is [available here](#).

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Microsoft, Windows, Windows XP, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.