# Installing PostgreSQL on Windows Server 2003

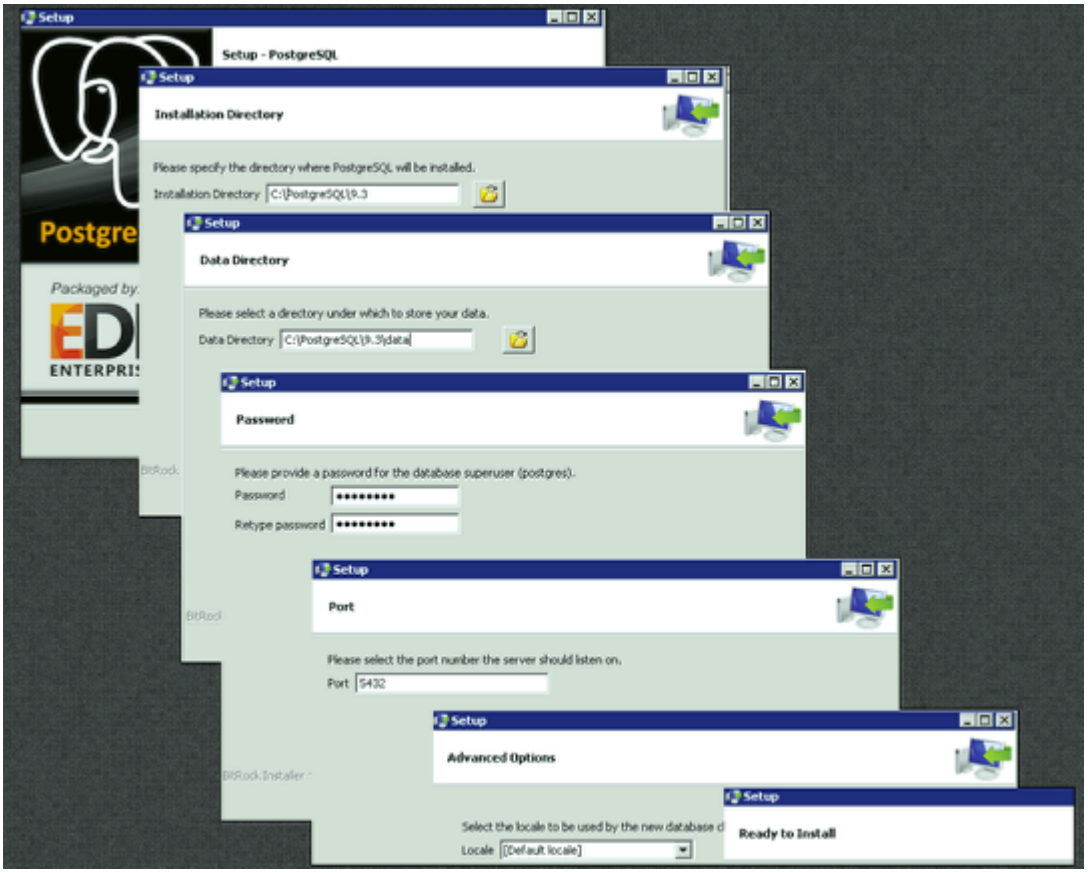
This Article covers installation of PostgreSQL Version 9.3 32-bit on Windows Server 2003 32-bit OS

## Preflight

| Item | Description | Windows Server 2003 R2 32-bit | 9.3 32-bit PostgreSQL Service Account | Username :postgres Password: postgres | C:\PostgreSQL\9.3 Data Directory | C:\PostgreSQL\9.3\data |
|---|---|---|---|---|---|---|

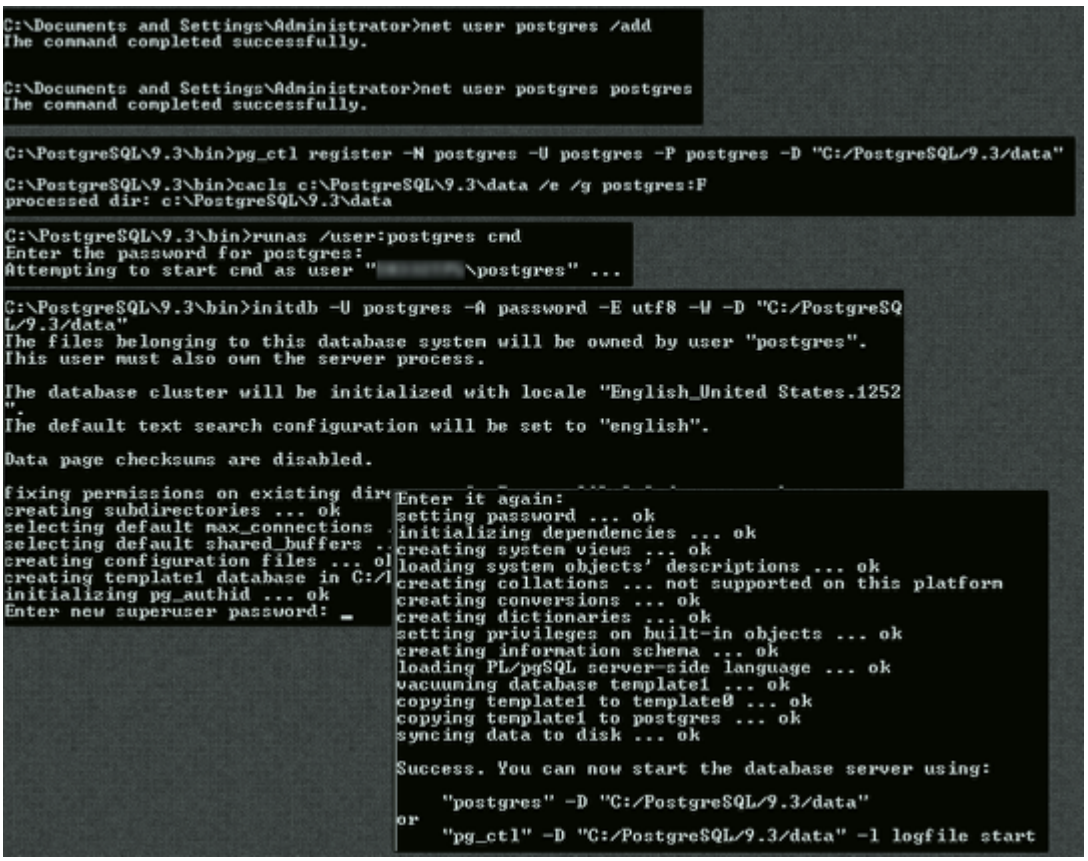
@!:{This article presents the steps from the command line (cmd) perspective.

## Launch PostgreSQL Installation



Installer Filename: postgresql-9.3.5-1-windows.exe
Launch the installer
Next • Data Directory: C:\PostgreSQL\9.3\data • Password: postgre • Port: 5432 • Locale: Default locale • Next …

## Create The PostgreSQL Service User & Initialize Database Instance




Create the PostgreSQL User Account
[shell]net user postgres /add[/shell]

Set the password for the User Account to ‘postgres’
[shell]net user postgres postgres[/shell]

Register the PostgreSQL Windows Service and set logon as the postgres user
[shell]pg_ctl register -N postgres -U postgres -P postgres -D "C:/PostgreSQL/9.3/data"[/shell]

Set appropriate permissions on the PostgreSQL data directory
[shell]cacls c:\PostgreSQL\9.3\data /e /g postgres:F[/shell]

Initialize the database instance, specifying the data directory from above
[shell]initdb -U postgres -A password -E utf8 -W -D "C:/PostgreSQL/9.3/data"[/shell]

Start the PostgreSQL Service
[shell]net start postgres[/shell]

This covers the installation and default configuration of PostgreSQL on Windows Server 2003 R2

## Post-Installation Notes

Ensure the postgreSQL bin folder is in Path environmental variable

[divider]

## Appendix

[divider]

## Troubleshooting

Troubleshooting Scenarios

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Error | Possible Cause | Possible Solution When you attempt starting the PostgreSQL service, you encounter an error similar to: "*The postgres service on Local Computer started and then stopped. Some services stop automatically if they have no work to do, for example, the Performance Logs and Alerts service.*" | The PostgreSQL Database Instance has not been initialized | Initialize the database instance then try starting the service again Additionally, you can verify that the PostgreSQL service account has adequate permissions on the data directory When you try executing any arbitrary PostgreSQL queries, you encounter an error similar to: "*Execution of PostgreSQL by a user with administrative permissions is not permitted. The server must be started under an unprivileged user ID to prevent possible system security compromises. See the documentation for more information on how to properly start the server.* | You registered the PostgreSQL service with an administrative user | De-register service [shell]pg_ctl.exe unregister -N postgres [/shell] Register service as a non-administrative user account (e.g. postgres) [shell]pg_ctl register -N postgres -U postgres -P postgres -D "C:/PostgreSQL/9.3/data"[/shell] You try to initialize the database instance but encount an error similar to: "*initdb: directory "C:/PostgreSQL/9.3/data" exists but is not empty If you want to create a new database system, either remove or empty the directory "C:/PostgreSQL/9.3/data" or run initdb with an argument other than "C:/PostgreSQL/9.3/data*". | You are trying to initialize the database instance, but the data directory is not empty | Delete the contents of the data directory and retry initializing the database instance Upon initializing the database instance, you encounter errors related to permissions, similar to: *fixing permissions on existing directory C:/PostgreSQL/9.3/data … initdb: could not change permissions of directory "C:/PostgreSQL/9.3/data": Permission denied* | The PostgreSQL service account does not have adequate permissions on the data directory | Grant full permissions on the data directory to the PostgreSQL service account, e.g.: [shell]cacls c:\PostgreSQL\9.3\data /e /g postgres:F[/shell] |

---

# Difference between the SET and SELECT statements when assigning variables in T-SQL

see:{http://vyaskn.tripod.com/differences_between_set_and_select.htm
see:{http://stackoverflow.com/questions/3945361/t-sql-set-versus-select-when-assigning-variables

## Quick Summary of Differences

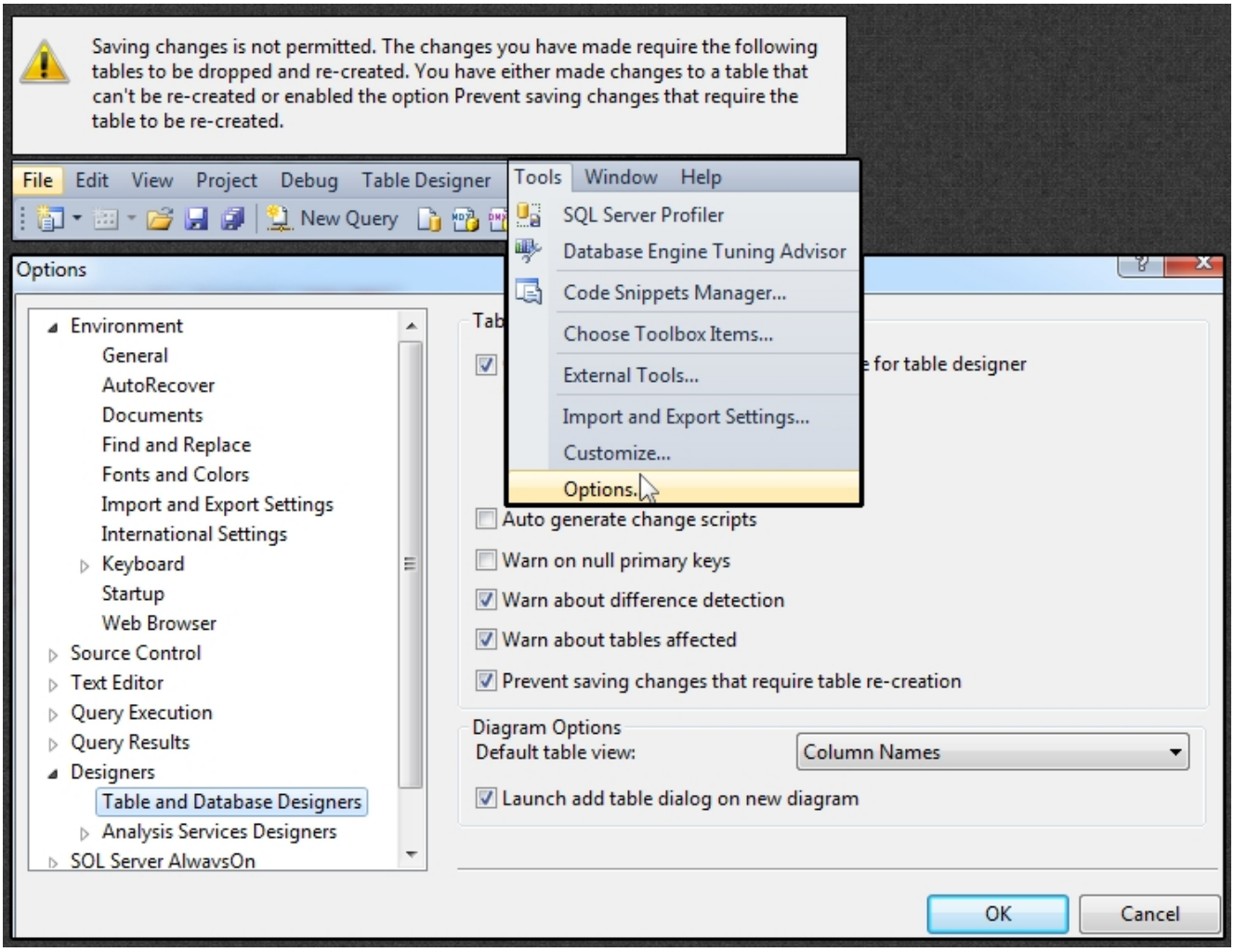
1. SET is the ANSI standard for variable assignment, SELECT

is not.
2. SET can only assign one variable at a time, SELECT can make multiple assignments at once.
3. If assigning from a query, SET can only assign a scalar value. If the query returns multiple values/rows then SET will raise an error. SELECT will assign one of the values to the variable and hide the fact that multiple values were returned (so you'd likely never know why something was going wrong elsewhere – have fun troubleshooting that one)
4. When assigning from a query if there is no value returned then SET will assign NULL, where SELECT will not make the assignment at all (so the variable will not be changed from it's previous value)
5. As far as speed differences – there are no direct differences between SET and SELECT. However SELECT's ability to make multiple assignments in one shot does give it a slight speed advantage over SET.

---

# SQL Server 2008 – Working With Tables

## Allow Saving Changes To Tables in SQL Server Management Studio




When attempting changes to Table structure via SQL Server Management Studio (SSMS), you may encounter an error:

“Saving changes is not permitted. The change you have made requires the following table to be dropped and re-created.
You have either made changes to a table that can’t be recreated or enabled the option prevent saving changes that require the table to be re-created.”

The fix is to disable the ‘Prevent saving changes that require table re-creation’ in the options dialog.

---

# SQL Server 2008 Troubleshooting Scenarios

In this article I cover troubleshooting scnenarios I've come across in my experience working with (not really managing) SQL Server 2008. ToDo: Add whatever notes yet undocumented, as well as any future scenarios

## Purging a bogus database from master.sys.databases

Scneario: I created a database refresh script. On accident, I submitted a refresh job to it where the target database name was that of the .bak file! The script produced an error and quit, but not before it created a database of that name. My coworker deleted it, but the darn thing remained in master.sys.databases. A simple drop database command was all I needed to clean this (small) mess:

[code language="sql" gutter="false"]

```
select * from master.sys.databases where name like '%.bak%'
drop database [D:\temp\database.bak]
```

[/code]