

For A2 ...

- ***If you do A2 on CDF***
 - <http://www.cdf.utoronto.ca/~csc343h/fall/postgres.shtml>
- ***If you do A2 on your own computer***
 - Download and install latest version of PostgreSQL
 - download: <http://www.postgresql.org/download/>
 - Installation Instruction: <http://www.postgresql.org/docs/8.3/interactive/installation.html>
 - Install on Windows: <http://pginstaller.projects.postgresql.org/>
 - Download and install JDBC driver
 - download: <http://jdbc.postgresql.org/download.html>
 - Installation instruction: <http://www.enterprisedb.com/docs/en/8.3/tools/jdbc/8.3.0/>

PostgreSQL

What?

- ❑ “The world’s most advanced open source database”
- ❑ Free 😊
- ❑ We’ll be using it for the 343 assignments.

Where?

- ❑ Central Site: [postgresql.org](https://www.postgresql.org)
- ❑ Latest Releases
 - 8.1 (8.1.4) (binary)
- ❑ Do not take an older version. It may need Cygwin.
- ❑ Pre-compiled binaries are available for Linux and Windows only.

Caveat Emptor

- ❑ Windows users: Only on NTFS
- ❑ Mac users: No native binaries are available
- ❑ But you can install by compiling the source. Google: postgresql on mac
- ❑ Linux 😊

Windows Install - Min Requirements

□ Min Requirements

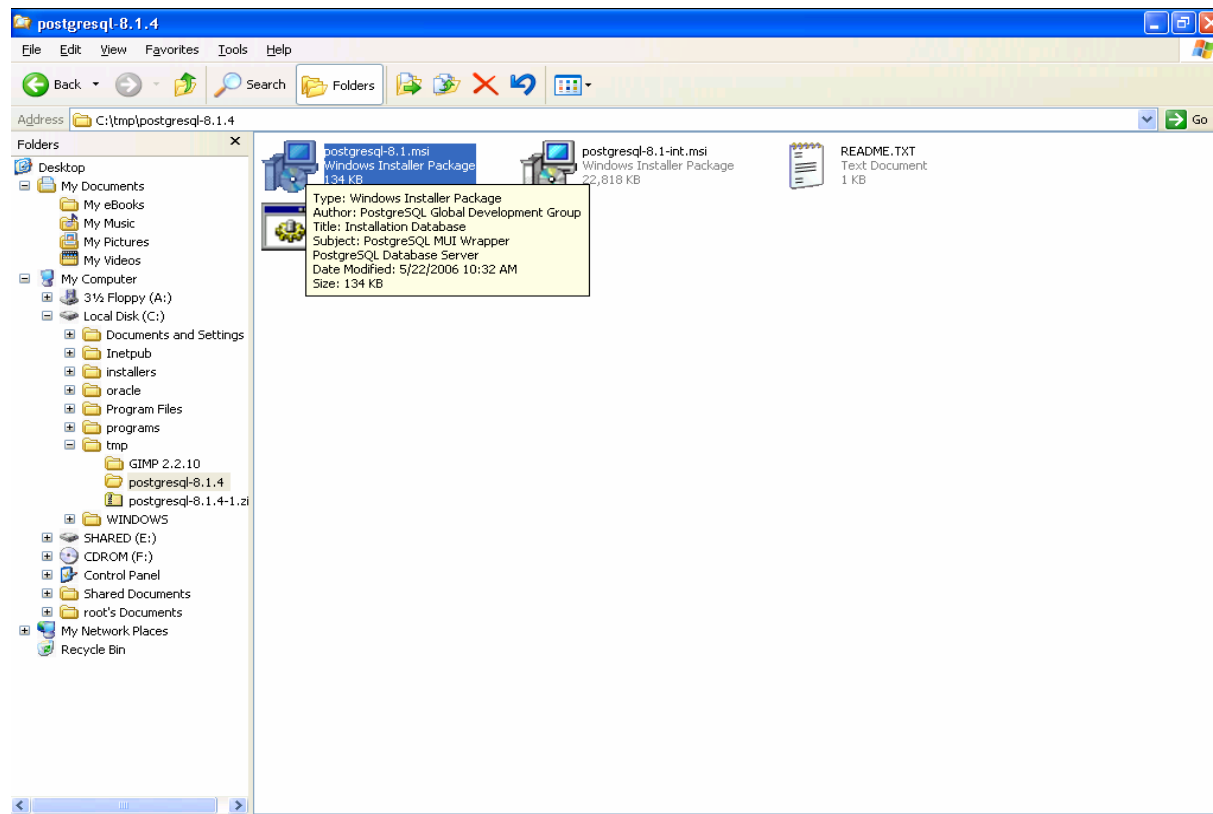
- CPU: 32-bit CPUs from either Intel or AMD
- Operating System: Windows XP or Windows Server 2003

□ Getting the Installer

- Download the latest version of PostgreSQL for Windows from the official website
 - site – <http://www.postgresql.org>

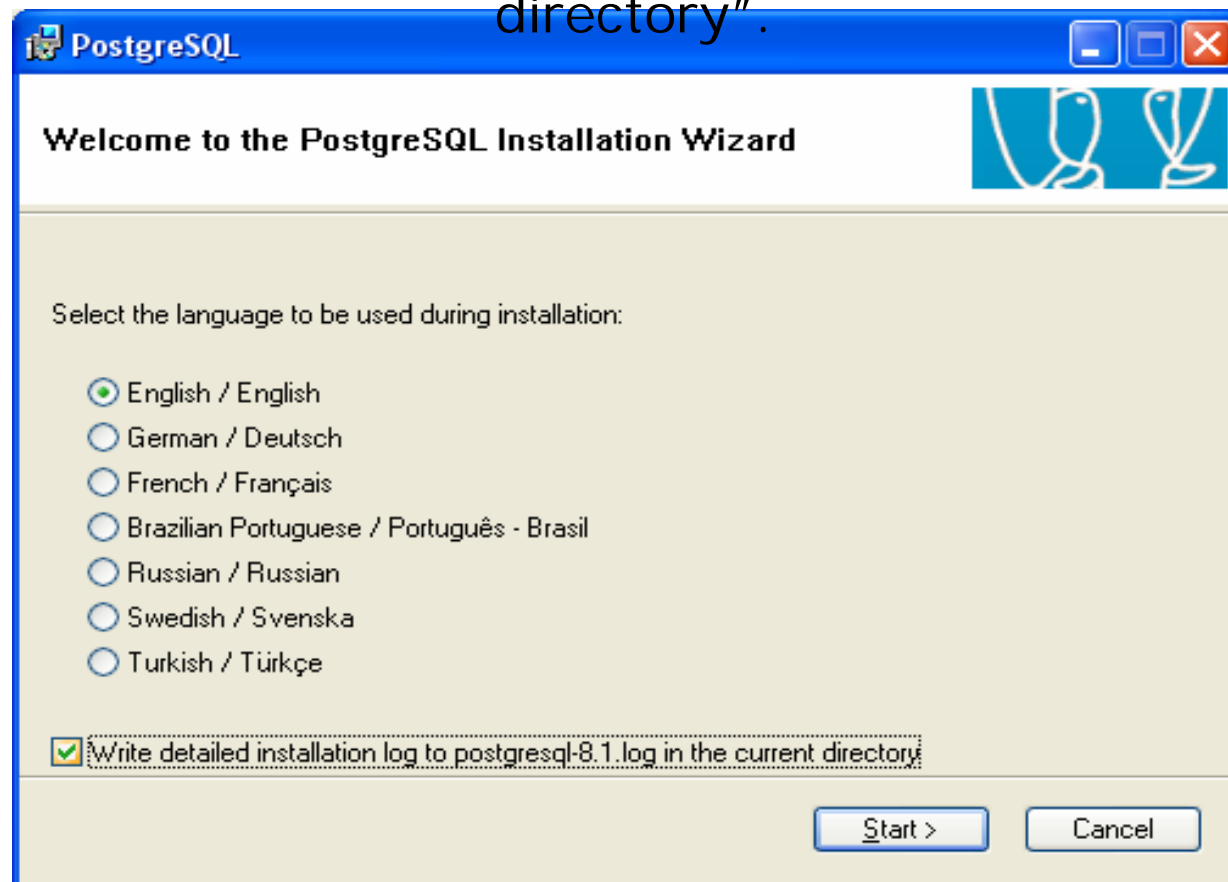
Installation

Installation comes with a zip file. Double-click on the postgresql-8.1.msi file to launch the installer.



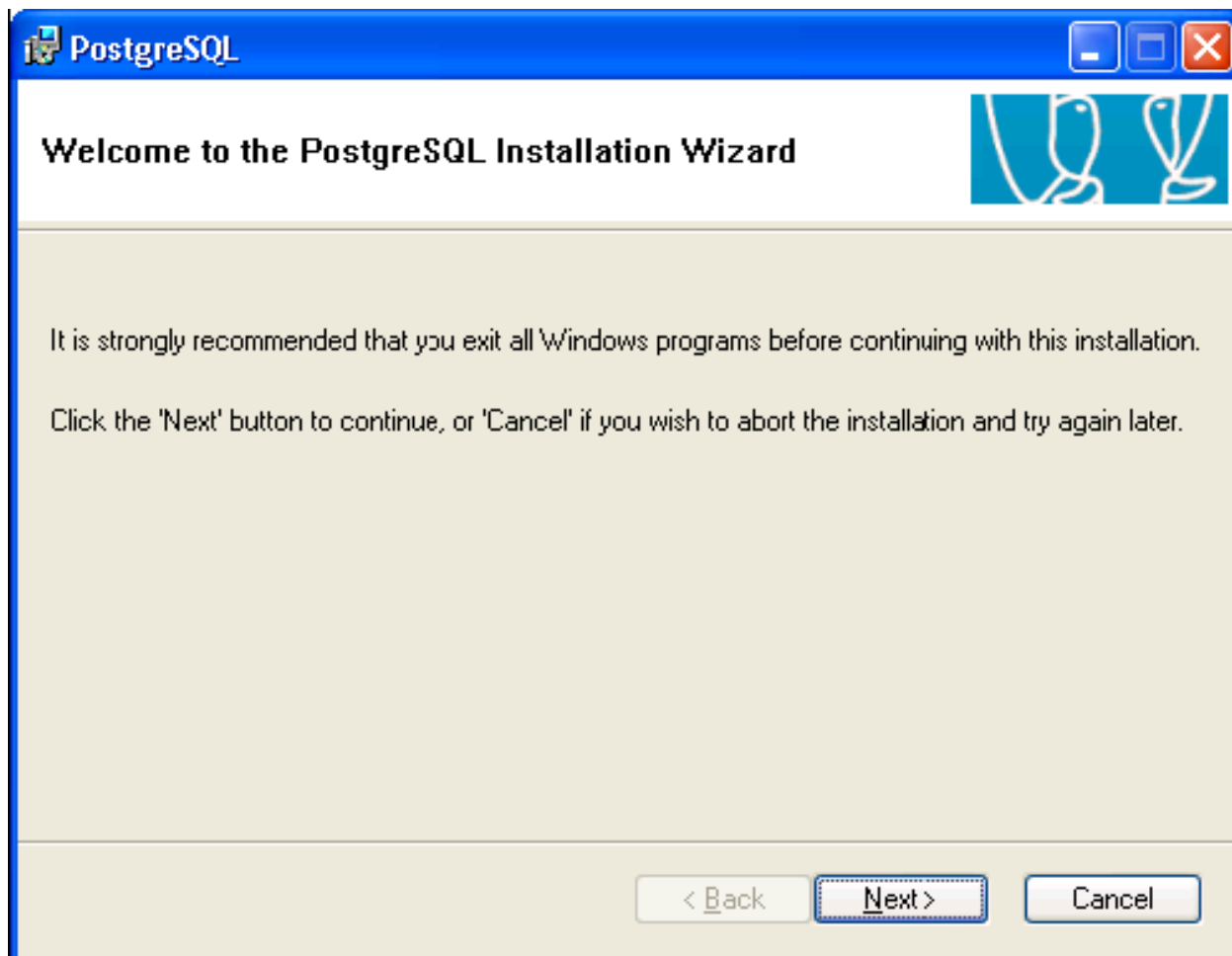
Installation

Check the “Write detailed installation log to postgresql-8.1.log in the current directory”.



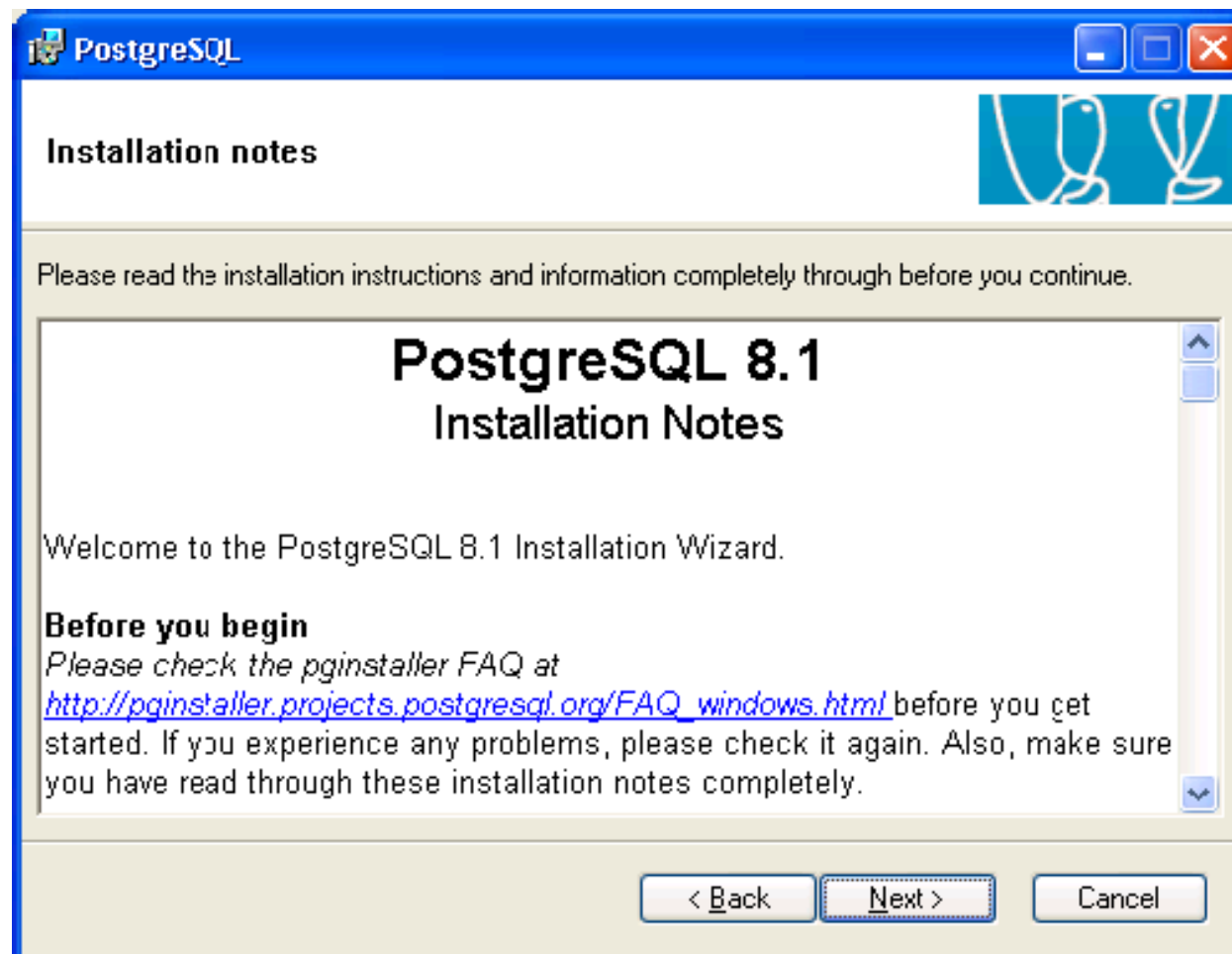
Installation

Click on Next



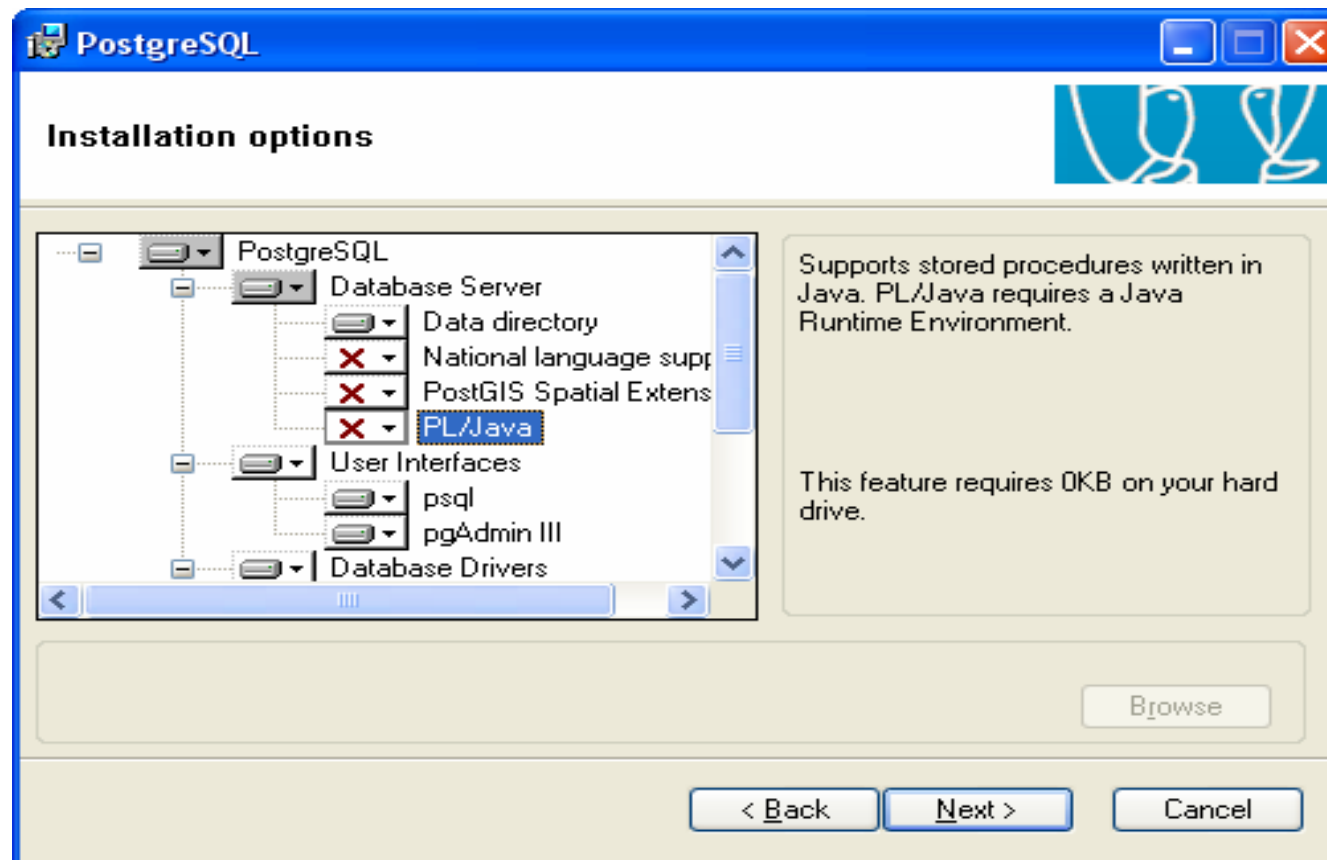
Installation

Read Installation notes and proceed to next screen



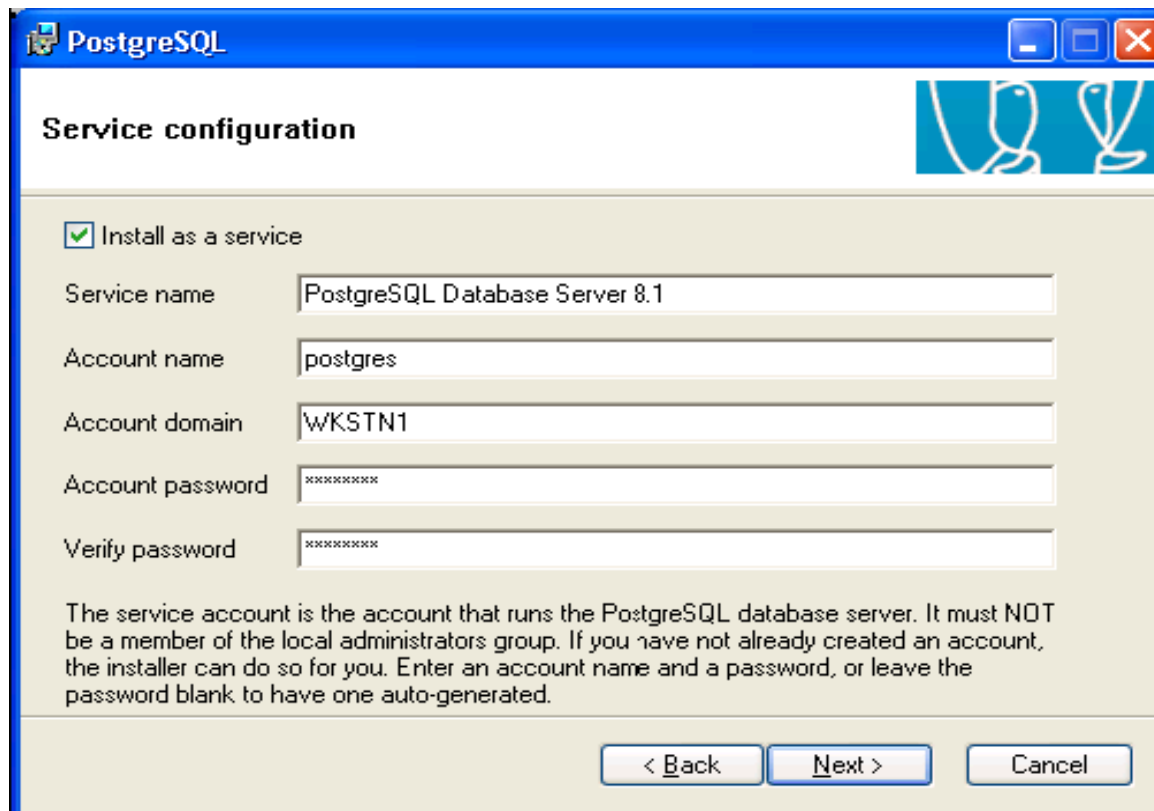
Installation

Select PL/Java: If you don't have Java Runtime, abort the installation and install Java runtime and start the install again
Change the installation directory (if needed)



Installation

- The “Account name” pertains to the Windows special user account that will be used to run the PostgreSQL database server.
- Make sure “Account domain” actually exists otherwise the installation will fail at a later point of time.



The screenshot shows the "Service configuration" dialog box for PostgreSQL. The window title is "PostgreSQL". The dialog has a blue header bar with the PostgreSQL logo and the text "Service configuration". Below the header, there is a checkbox labeled "Install as a service" which is checked. The main area contains five input fields: "Service name" (PostgreSQL Database Server 8.1), "Account name" (postgres), "Account domain" (WKSTN1), "Account password" (masked with asterisks), and "Verify password" (masked with asterisks). At the bottom, there is a paragraph of text explaining the service account requirements, and three buttons: "< Back", "Next >", and "Cancel".

Install as a service

Service name: PostgreSQL Database Server 8.1

Account name: postgres

Account domain: WKSTN1

Account password: *****

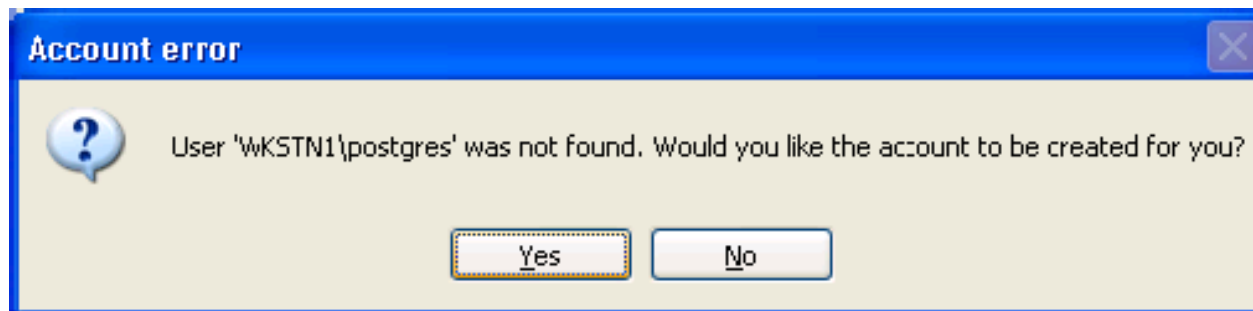
Verify password: *****

The service account is the account that runs the PostgreSQL database server. It must NOT be a member of the local administrators group. If you have not already created an account, the installer can do so for you. Enter an account name and a password, or leave the password blank to have one auto-generated.

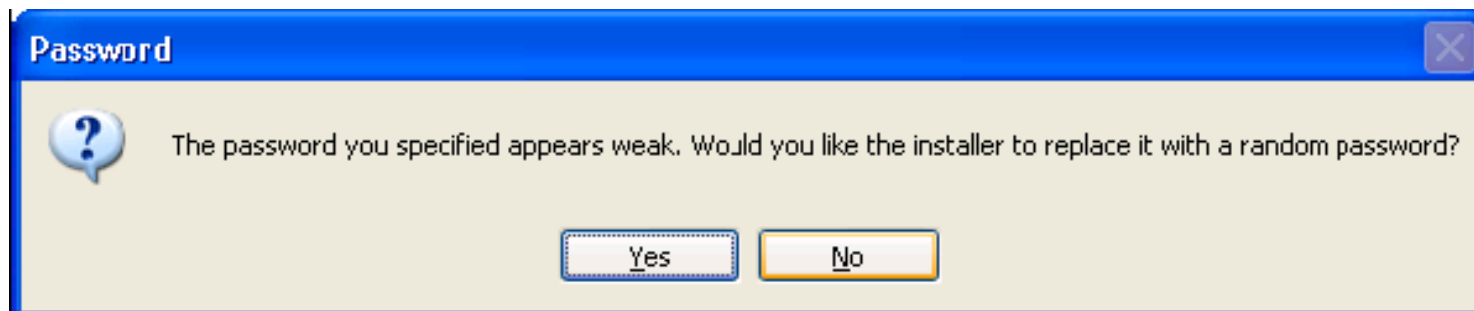
< Back Next > Cancel

Installation

Click on Yes. Installation will automatically create the account if that account doesn't exist.

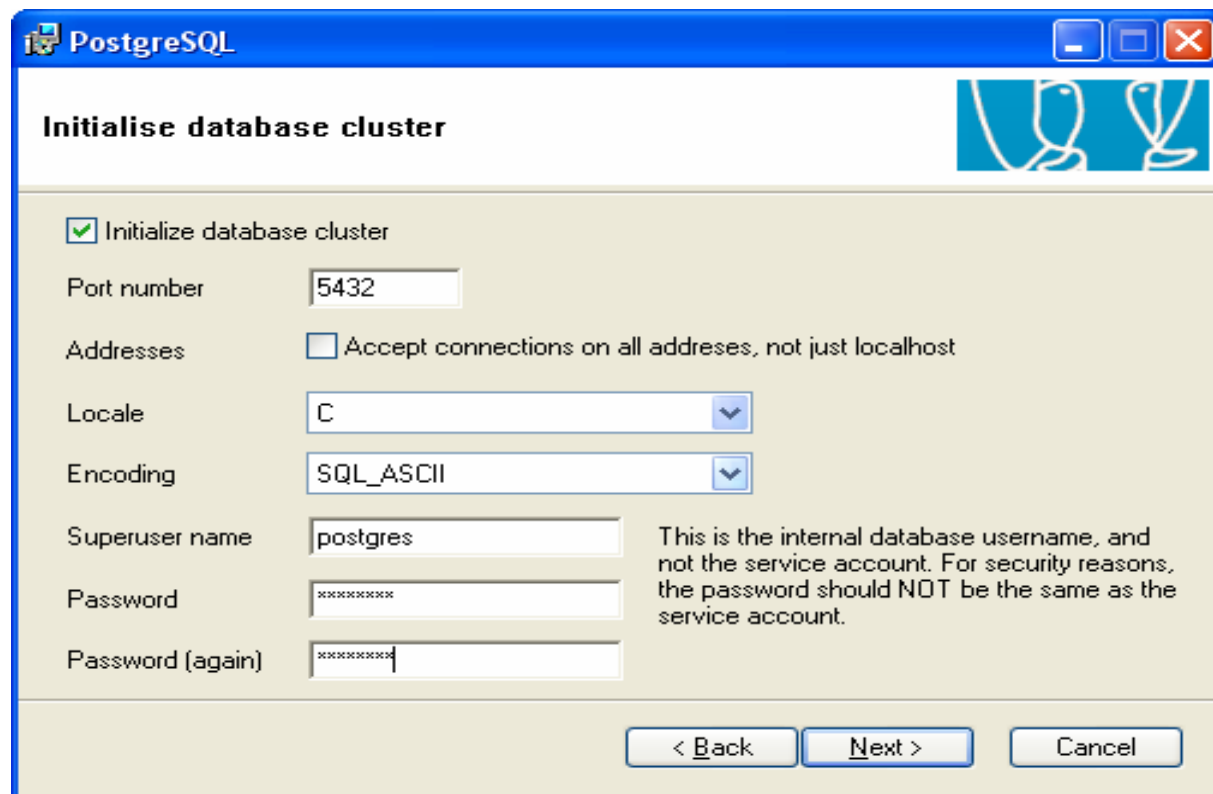


Select Yes/No depending on your choice. If yes, random password will be generated. If No, weak password will remain as your password.



Installation

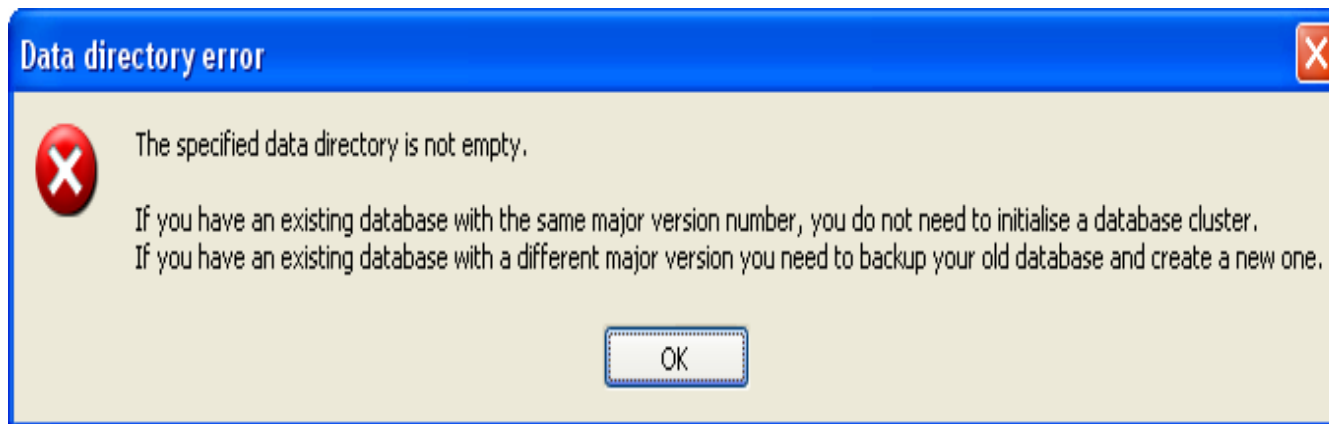
1. Do not confuse this “superuser name” with the Windows special user account created earlier.
2. The superuser here pertains to the PostgreSQL database server account that can create databases and roles and has unrestricted access whereas windows special user can be found in My computer->Local Users and Groups.



The screenshot shows the 'Initialise database cluster' dialog box in PostgreSQL. The window title is 'PostgreSQL'. The dialog has a blue header bar with the PostgreSQL logo and the text 'Initialise database cluster'. Below the header, there is a checkbox labeled 'Initialize database cluster' which is checked. The 'Port number' is set to '5432'. The 'Addresses' section has a checkbox 'Accept connections on all addresses, not just localhost' which is unchecked. The 'Locale' is set to 'C' and the 'Encoding' is set to 'SQL_ASCII'. The 'Superuser name' is 'postgres'. The 'Password' and 'Password (again)' fields are masked with asterisks. A note on the right side of the dialog states: '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.' At the bottom of the dialog, there are three buttons: '< Back', 'Next >', and 'Cancel'.

Install: Special Case (slide 13 cont.)

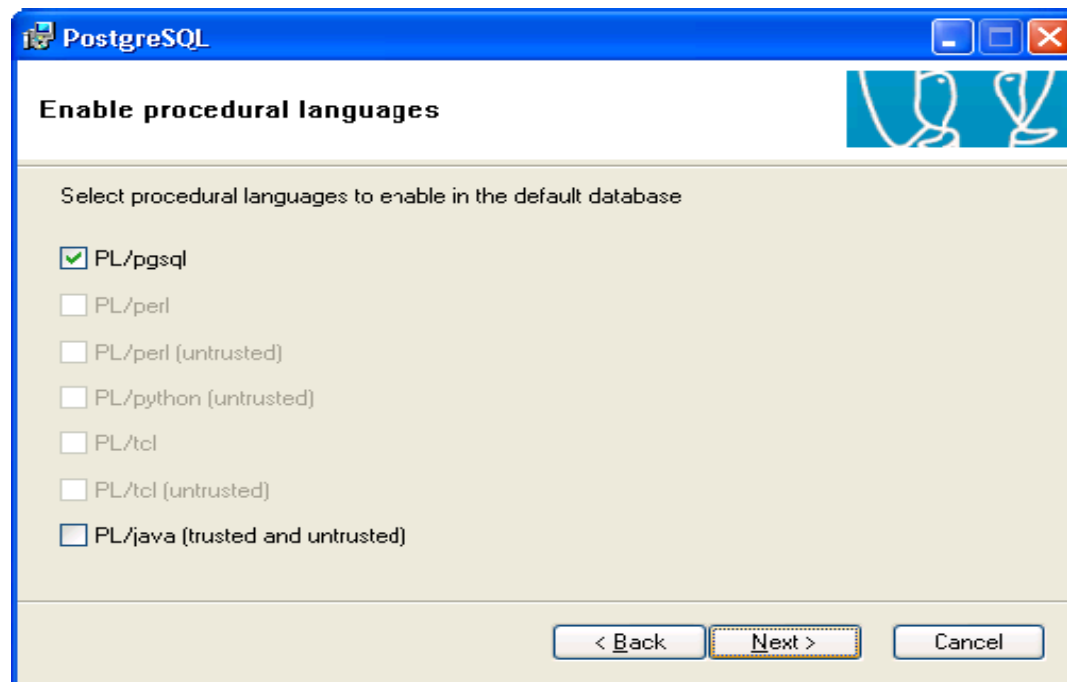
If you have already created some database using PostgreSQL, you do not need to initialize database cluster again.



Uncheck the initial database cluster option from the previous screen to avoid this error.

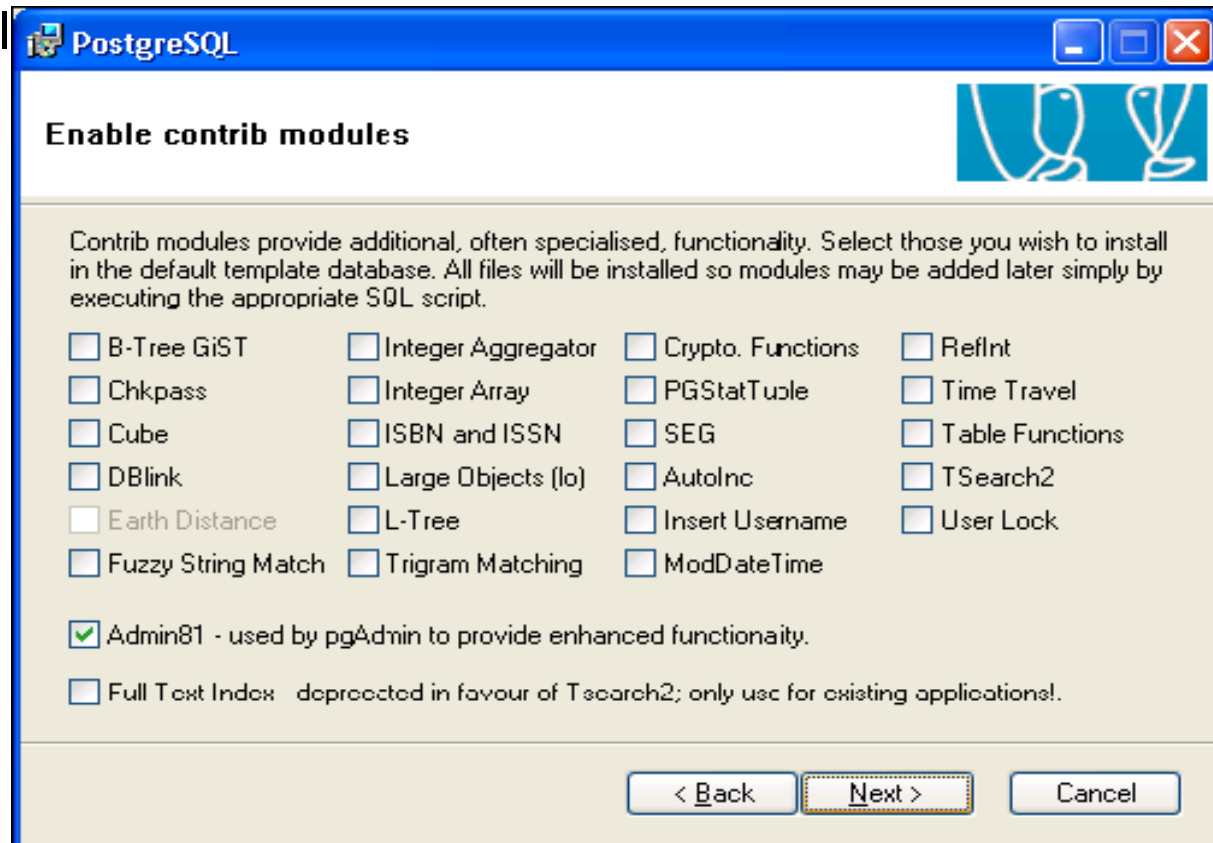
Installation

Select PL/pgsql and continue



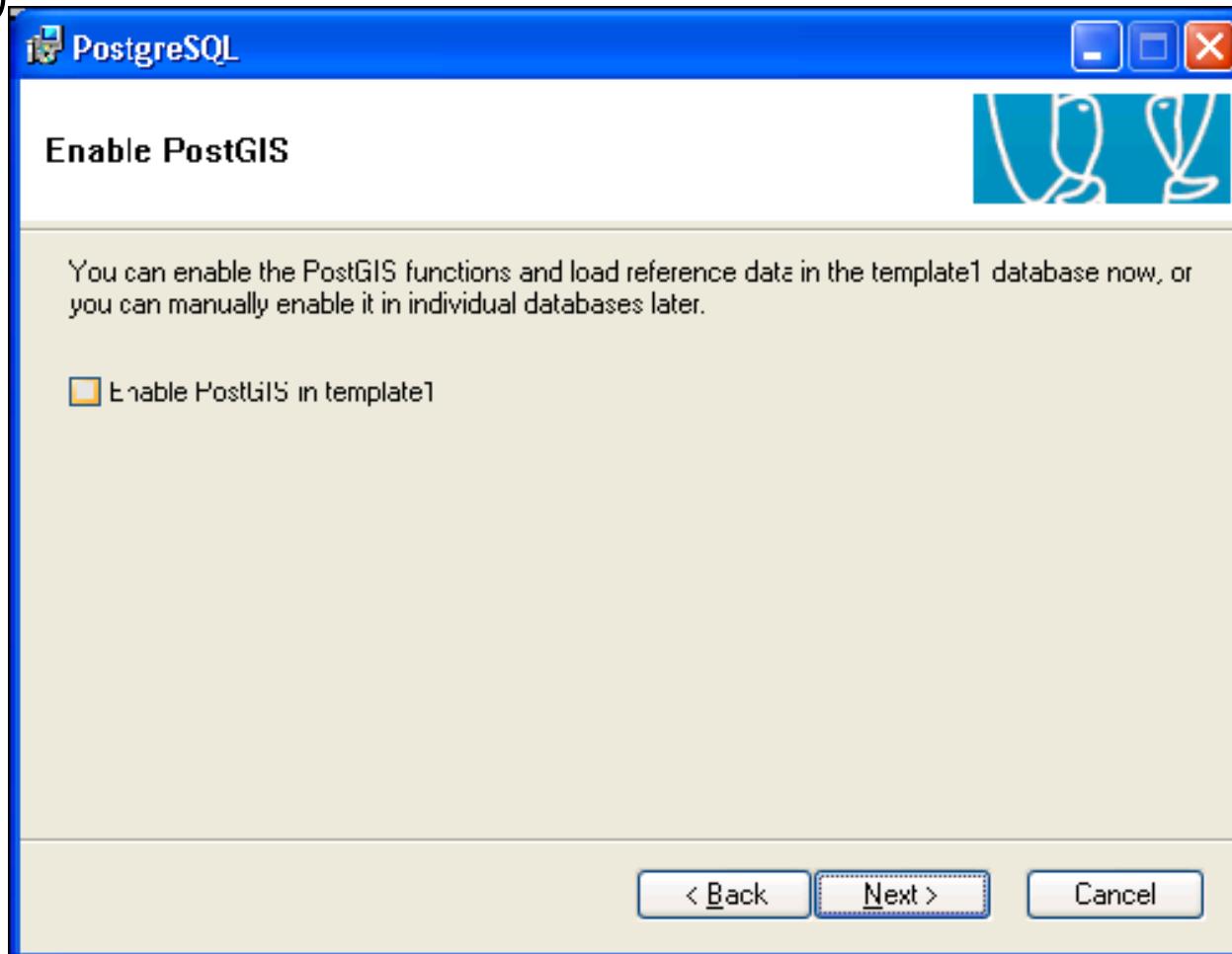
Installation

- Do NOT enable contrib modules in the default template database.
- For e.g. If crypto functions is enabled, every database that is created from default template will have crypto functions enabled.



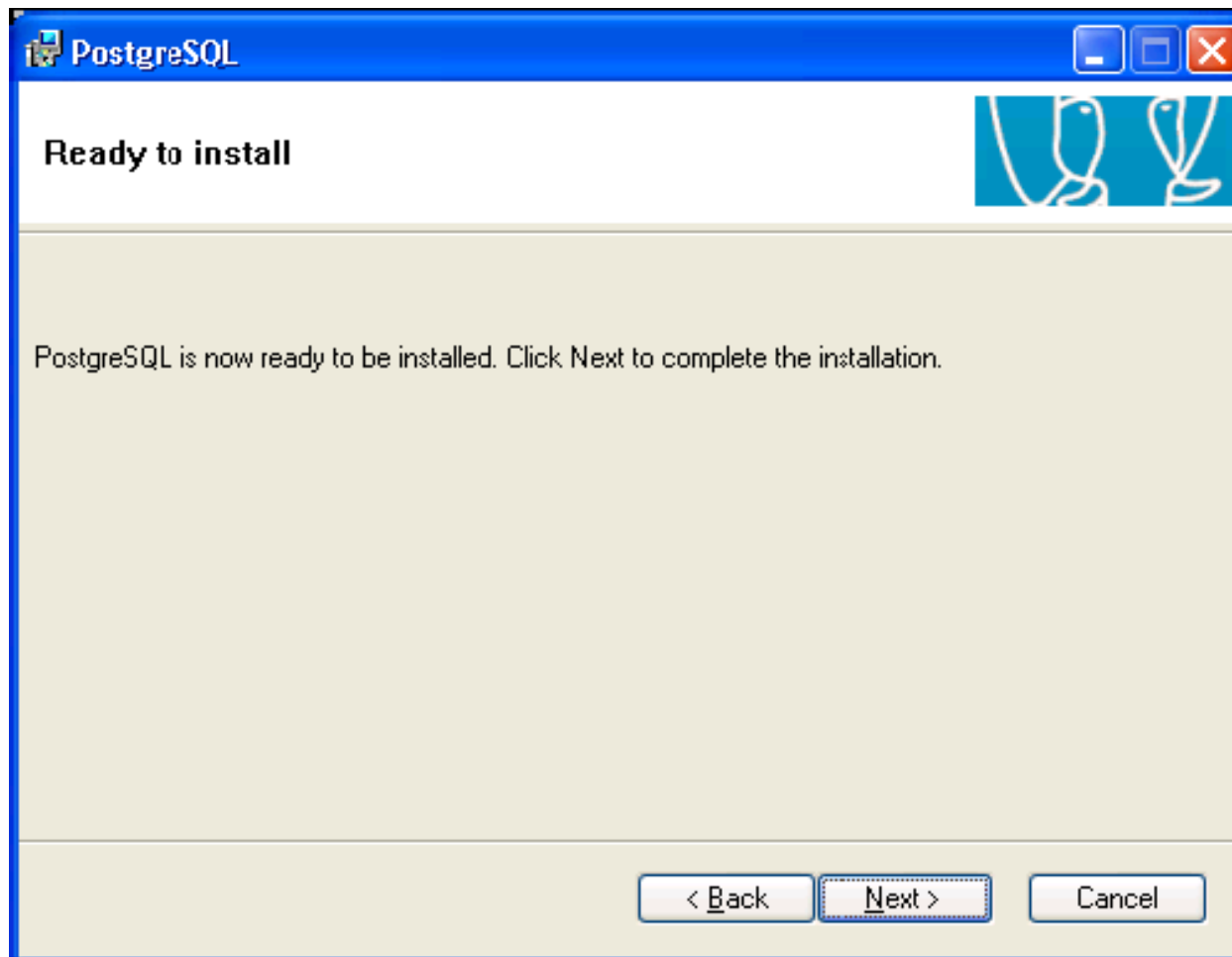
Installation

Don't check on PostGIS. By not checking, PostGIS functions from template1 are disabled. (which we don't want)



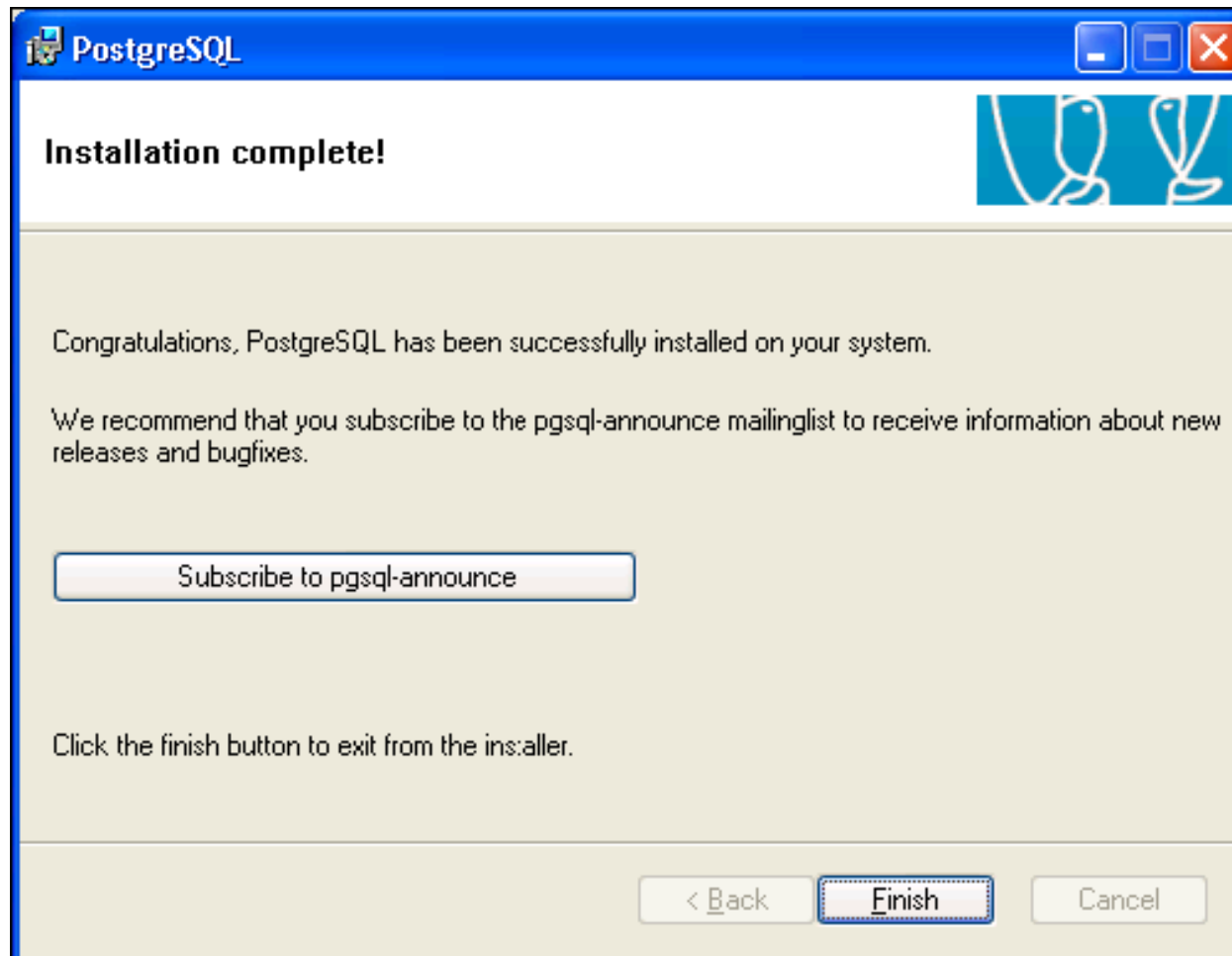
Installation

Click Next



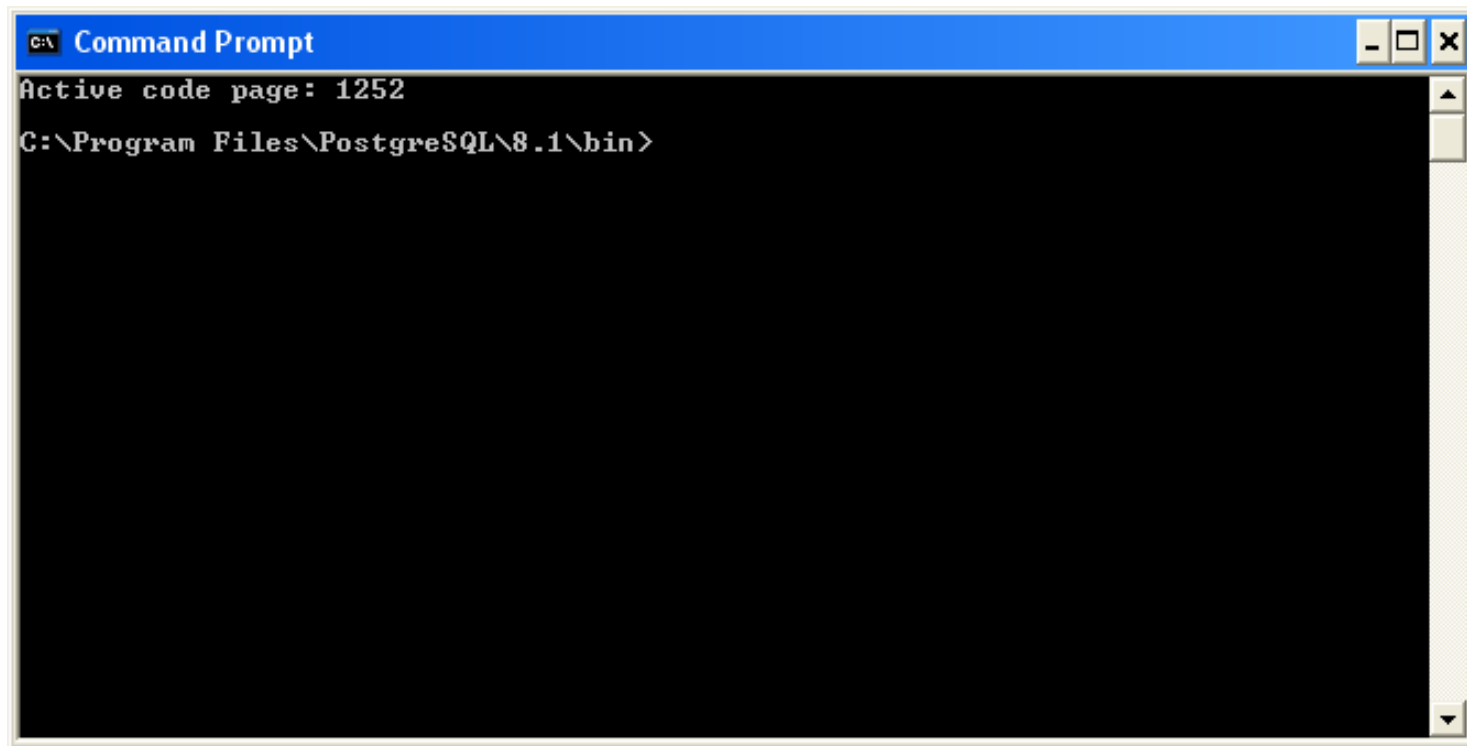
Installation

Click “Finish” and that's it for the installation.



Launch PostGRESQL command prompt

- 1) Run the PostgreSQL command prompt via Start -> All Programs -> PostgreSQL 8.1 -> Command Prompt. A Windows command prompt will appear.
- 2) psql is a command-line interface to PostgreSQL



```
Command Prompt
Active code page: 1252
C:\Program Files\PostgreSQL\8.1\bin>
```

How to connect to a PostgreSQL Server?

- ❑ The postgres database account is a superuser by default.
- ❑ Steps:
 1. Launch POSTGRESQL command prompt
 2. Enter the following at the command prompt
 - *C:\Program Files\PostgreSQL\8.1\bin>psql -U postgres -h localhost*
 3. Provide superuser “postgres” password
 4. -U postgres – indicates user name, -h localhost – indicates server is on local host
 5. PostgreSQL prompt will appear – means that we are connected to the database named “postgres”, the default database.
 - *postgres=#*

Setting a Sample Database

Setting up database

- Four steps:
 1. Create database owner
 2. Create storage for default table space and tighten security(if required)
 3. Create database
 4. Create database objects

Create Database owner

- ❑ Connect to default postgres database(Refer to slides 21, 22)
- ❑ To create a role, Enter:
 - `postgres=# CREATE ROLE <db_owner> LOGIN
PASSWORD '<sample_pwd>';`
- ❑ To verify creation of owner, Enter:
 - `postgres=# \du <db_owner>`

Create the default tablespace

- ❑ Use c:\pgdata folder for storing default table space
- ❑ If you are running POSTGRESQL server in a multiuser environment, then you need to tighten the security for the C:\pgdata folder. For desktop environment this is not necessary.
- ❑ To create the "sample_ts" tablespace, enter:
 - `postgres=# CREATE TABLESPACE sample_ts OWNER <db_owner> LOCATION 'c:/pgdata/sampledbs/system';`
- ❑ To verify table space creation, run
 - `postgres=# \db+ <sample_db>`

Create Database

- ❑ To create the "sample_db" database, enter:

- `postgres=# CREATE DATABASE <sample_db> OWNER <db_owner> TEMPLATE template0 TABLESPACE sample_ts;`

- ❑ To list all installed databases, enter:

- `postgres=# \l+`

Create Database objects

□ Steps involved

1. Connect to sample database and then enter

```
postgres=# \c <sample_db_name>
```

2. Run the SQL DB scripts or queries to create/update database objects.

3. List database objects:

```
postgres=# \d+
```

□ Finally, enter:

■ **postgres=# Analyze**

□ to update the statistics used by the PostgreSQL query planner to generate good execution plans for queries

JDBC Connectivity

- ❑ Install JDBC Driver
- ❑ Testing Driver
- ❑ Using JDBC Driver
- ❑ Simple examples

Install JDBC Driver

- ❑ Make sure to obtain the appropriate JDBC version
- ❑ Download the appropriate .jar file(s) into your machine for installing JDBC driver
- ❑ Set the class path
 - Add the complete path including the .jar file name to the JAVA CLASSPATH variable
 - OR provide class path as command line argument every time you run the Java programs.

Setting CLASSPATH (more details)

□ Two Methods:

■ Set CLASSPATH environment variable

□ Only for current command prompt session, run

CMD> Set CLASSPATH=C:\tmp\psql-driver.jar

□ To set it CLASSPATH permanently

Open Control Panel -> system and add a new environment variable called CLASSPATH

■ Provide CLASSPATH for each program you run

□ **CMD> java -classpath "c:\tmp\psql-driver.jar" abc.java**

Testing Driver

- ❑ To test if driver passes through the class loader, lookup by class name, as shown in the Java code snippet.

Example: Class name lookup

```
try {  
    Class.forName("org.postgresql.Driver");  
} catch (ClassNotFoundException cnfe) {  
    System.err.println("Couldn't find driver  
class:");  
    cnfe.printStackTrace();  
}
```

Using JDBC Driver (steps below)

1. Importing JDBC

- Any source that uses JDBC needs to import `JDBC.sql.*` packages

2. Load the driver

- `Class.forName("org.postgresql.Driver");`
- This will automatically register itself with JDBC driver

3. Connecting to database, enter

- `Connection db = DriverManager.getConnection(url, username, password);`
- For e.g. URL may look like `jdbc:[drivertype]:[database]`

4. Closing the connection

- `db.close();`

Simple JDBC connection example

```
import java.sql.DriverManager;
import java.sql.Connection;
import java.sql.SQLException;

public class Example1 {
    public static void main(String[] argv) {
        System.out.println("Checking if Driver is registered with
            DriverManager.");

        try {
            Class.forName("org.postgresql.Driver");
        } catch (ClassNotFoundException cnfe) {
            System.out.println("Couldn't find the driver!");
            System.out.println("Let's print a stack trace, and exit.");
            cnfe.printStackTrace();
            System.exit(1);
        }
    }
}
```

Simple JDBC Connection example (Cont)

```
System.out.println("Registered the driver ok, so let's make a connection.");
Connection c = null;
    try {
        // The second and third arguments are the username and password,
        // respectively. They should be whatever is necessary to connect
        // to the database.
        c = DriverManager.getConnection("jdbc:postgresql://localhost/booktown",
                                       "username", "password");
    } catch (SQLException se) {
        System.out.println("Couldn't connect: print out a stack trace and exit.");
        se.printStackTrace();
        System.exit(1);
    }
    if (c != null)
        System.out.println("Hooray! We connected to the database!");
    else
        System.out.println("We should never get here.");
}
```

Simple JDBC Select

```
Statement s = null;
try {
    s = c.createStatement();
} catch (SQLException se) {
    System.out.println("We got an exception while creating a statement:" +
        "that probably means we're no longer connected.");
    se.printStackTrace();
    System.exit(1);
}
ResultSet rs = null;
try {
    rs = s.executeQuery("SELECT * FROM books");
} catch (SQLException se) {
    System.out.println("We got an exception while executing our query:" +
        "that probably means our SQL is invalid");
    se.printStackTrace();
    System.exit(1);
}
int index = 0;
try {
    while (rs.next()) {
        System.out.println("Here's the result of row " + index++ + ":");
        System.out.println(rs.getString(1));
    }
} catch (SQLException se) {
    System.out.println("We got an exception while getting a result:this " +
        "shouldn't happen: we've done something really bad.");
    se.printStackTrace();
    System.exit(1);
}
```

Some Errors

- ❑ No driver available SQLException being thrown while opening connection:
 - driver path might not be specified in the class path, or the value in the parameter is correct.
- ❑ Might throw ClassNotFoundException if driver is not installed

Questions?

References

- ❑ Central site: www.postgresql.org
 - Contains documentation, latest releases, FAQ and lots of other stuff
- ❑ Windows Installation reference document
 - http://www.charltonlopez.com/documents/getting_started_with_postgresql_for_windows.zip
- ❑ Windows installation screenshots from PostgreSQL site:
 - <http://pginstaller.projects.postgresql.org/>
- ❑ Windows installation FAQ:
 - http://pginstaller.projects.postgresql.org/faq/FAQ_windows.html
- ❑ Version 8.1.4
 - Doc Reference: <http://www.postgresql.org/docs/8.1/static/index.html>
 - Tutorial:
<http://www.postgresql.org/docs/8.1/static/tutorial.html>
 - SQL Reference:
<http://www.postgresql.org/docs/8.1/static/sql.html>
- ❑ JDBC:
 - <http://www.cs.toronto.edu/~faye/343/f07/postgres.shtml>
- ❑ General FAQ:
 - <http://www.postgresql.org/docs/faqs.FAQ.html>