



PostGIS

Release 3.0.1

PostGIS User's Guide

May 28, 2020

Contents

1	Installing PostGIS	2
1.1	Using an RPM Package to Install PostGIS	2
1.2	Using StackBuilder Plus to Install PostGIS	4
2	Creating Extensions	9
3	Using PostGIS	11
4	Upgrading PostGIS	13
5	Uninstalling PostGIS	14
6	Conclusion	16
	Index	17

PostGIS is a PostgreSQL extension that allows you to store Geographic Information Systems (GIS) objects in an Advanced Server database. PostGIS includes functions for analyzing and processing GIS objects, and support for GiST-based R-Tree spatial indexes.

The complete documentation for PostGIS is added to your system during the installation, and is available online at the [PostGIS project site](#).

CHAPTER 1

Installing PostGIS

The following table lists the latest PostGIS versions and their corresponding Advanced Server versions.

The PostGIS version required by your Advanced Server installation is version-specific, but the documented and supported functionality of each version is the same. The information in this guide applies to each version listed in the table below.

PostGIS Version	Supported Advanced Server Versions
PostGIS 3.0.1	Advanced Server 12
PostGIS 2.5.4	Advanced Server 11 and 12
PostGIS 2.5.3	Advanced Server 12
PostGIS 2.5.1	Advanced Server 11
PostGIS 2.4.6	Advanced Server 9.6, 10 and 11
PostGIS 2.3.8	Advanced Server 9.6 and 10
PostGIS 2.1.9	Advanced Server 9.5

To view a complete list of EnterpriseDB supported platforms, visit [the EnterpriseDB website](#).

1.1 Using an RPM Package to Install PostGIS

You must install Advanced Server before installing PostGIS. For details about installing and configuring Advanced Server, see [the EDB Postgres Advanced Server Installation Guide](#). If you have previously used an RPM package to install Advanced Server, you have probably already created the repository configuration file and have EnterpriseDB credentials. If that is the case, you can simply install PostGIS with the command:

```
yum -y install edb-as<xx>-postgis-<y.y.y>
```

Where <xx> is the Advanced Server version and <y.y.y> is the PostGIS version you want to install. For example, to install PostGIS 3.0.1 for Advanced Server 12, execute the following command:

```
yum -y install edb-as12-postgis-3.0.1
```

If you have removed your EnterpriseDB repository configuration file, you can use the steps in the following section to recreate it.

Creating a Repository Configuration File

The following steps provide detailed information about adding a repository configuration file to your system and installing PostGIS from an RPM package.

1. Use the following command to create the repository configuration file:

```
yum -y install https://yum.enterprisedb.com/edb-repo-rpms/edb-repo-latest.noarch.rpm
```

2. The repository configuration file is named `edb.repo`; the file resides in `/etc/yum.repos.d`. Use your choice of editor to ensure that the value of the `enabled` parameter is 1, and replace the `<username>` and `<password>` placeholders in the `baseurl` specification with the name and password of a registered EnterpriseDB user.

```
[edb]
name=EnterpriseDB RPMs $releasever - $basearch
baseurl=https://<username>:<password>@yum.enterprisedb.com/edb/redhat/rhel-
$releasever-$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/ENTERPRISEDB-GPG-KEY
```

If you need EnterpriseDB credentials, [visit the EnterpriseDB website](#).

3. After saving your changes to the configuration file, you can use the `yum install` command to install PostGIS:

```
yum install edb-as<xx>-postgis-<y.y.y>
```

Where `<xx>` is the Advanced Server version and `<y.y.y>` is the PostGIS version you want to install.

For example, to install PostGIS 3.0.1 on Advanced Server 12, execute the following command:

```
yum install edb-as12-postgis-3.0.1
```

When you install an RPM package that is signed by a source that is not recognized by your system, yum may ask for your permission to import the key to your local server. If prompted, and you are satisfied that the packages come from a trustworthy source, enter `y` and press Return to continue.

The `edb-as<xx>-postgis-<y.y.y>` package installs the following PostGIS component packages:

```
edb-as<xx>-postgis-<y.y.y>
edb-as<xx>-postgis-utils
edb-as<xx>-postgis-docs
edb-as<xx>-postgis-core
edb-as<xx>-postgis-jdbc
```

After installing PostGIS with a package manager, please note that you must manually create a template database and the required PostGIS extension.

1.2 Using StackBuilder Plus to Install PostGIS

You must install Advanced Server before installing PostGIS. If you have used the graphical Setup wizard to install Advanced Server, you can use StackBuilder Plus to add PostGIS to your installation. For details about using the graphical installer to install and configure Advanced Server, see the [EDB Postgres Advanced Server Installation Guide for Windows](#).

1. Open StackBuilder Plus and select your Advanced Server installation from the drop-down list on the `Welcome` window. Click `Next` to continue to the application selection page.

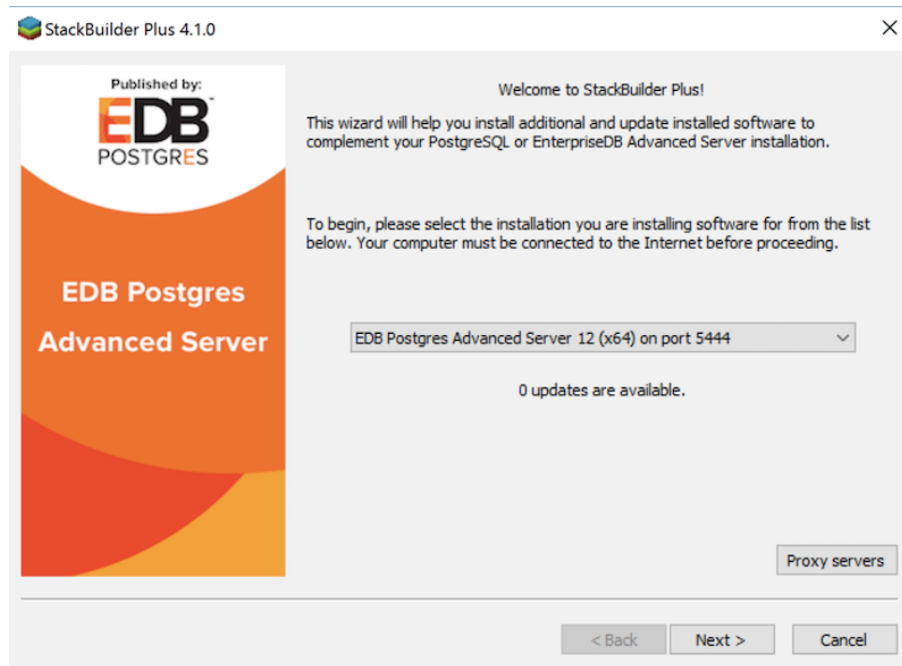


Fig. 1: *The StackBuilder Plus Welcome window*

2. Expand the `Spatial Extensions` node, and check the box next to the PostGIS version. Click `Next` to continue.
3. The selected packages and the default download directory where the package will be installed are displayed; change the locations if required. Click `Next`.
4. Once you have downloaded the installation files, a confirmation message is displayed. Click `Next` to start the PostGIS installation.

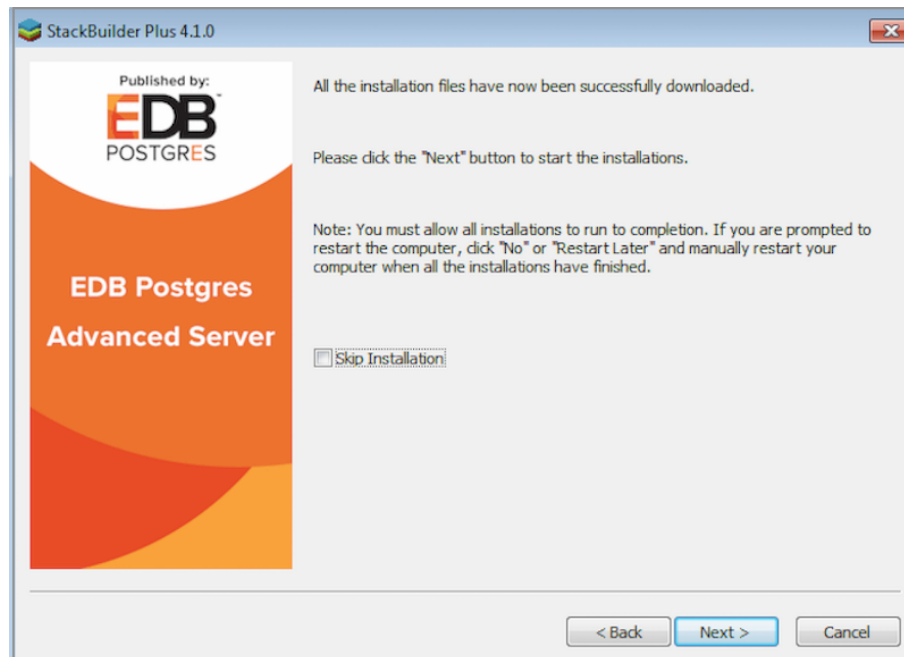


Fig. 2: *installing_postgis.png*

5. Select an installation language and click OK.

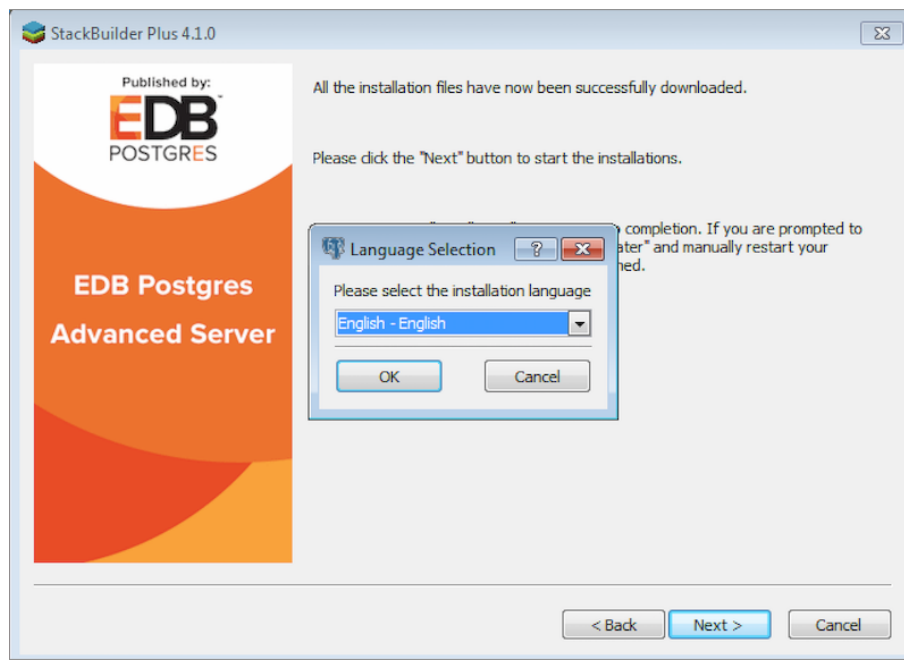


Fig. 3: *The language selection window*

6. The PostGIS welcome screen is displayed. Click Next.

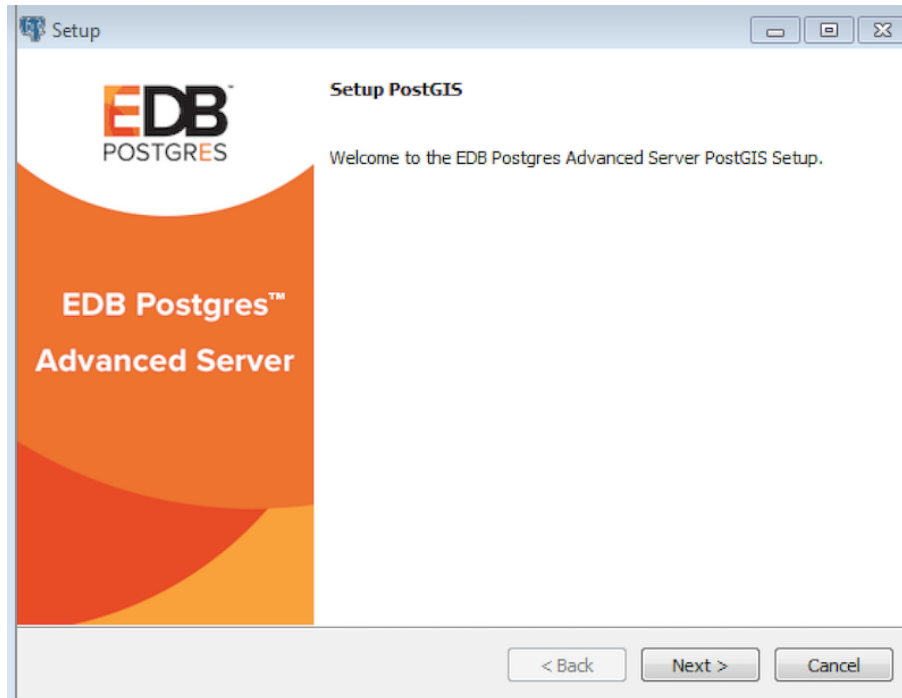


Fig. 4: *The Postgis welcome window*

7. Use the Installation Directory field to specify the directory in which you wish to install the PostGIS software. Click Next to continue.

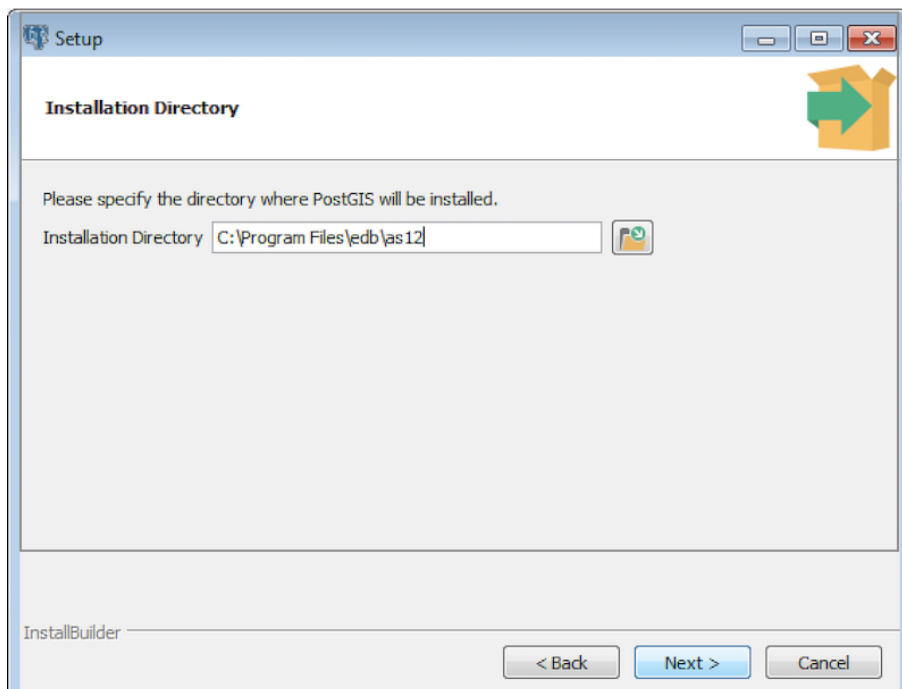


Fig. 5: *The Installation Directory window*

8. Use fields on the EDB Postgres Advanced Server Installation Details window to provide connection information for the Advanced Server host:

- Use the `Host` field to identify the system on which Advanced Server resides.
- Provide the name of the role that PostGIS will use for connections to the server in the `User Name` field.
- Provide the password associated with the role in the `Password` field.
- Use the `Port` field to identify the listener port that Advanced Server monitors for client connections.

Then, click `Next` to continue.

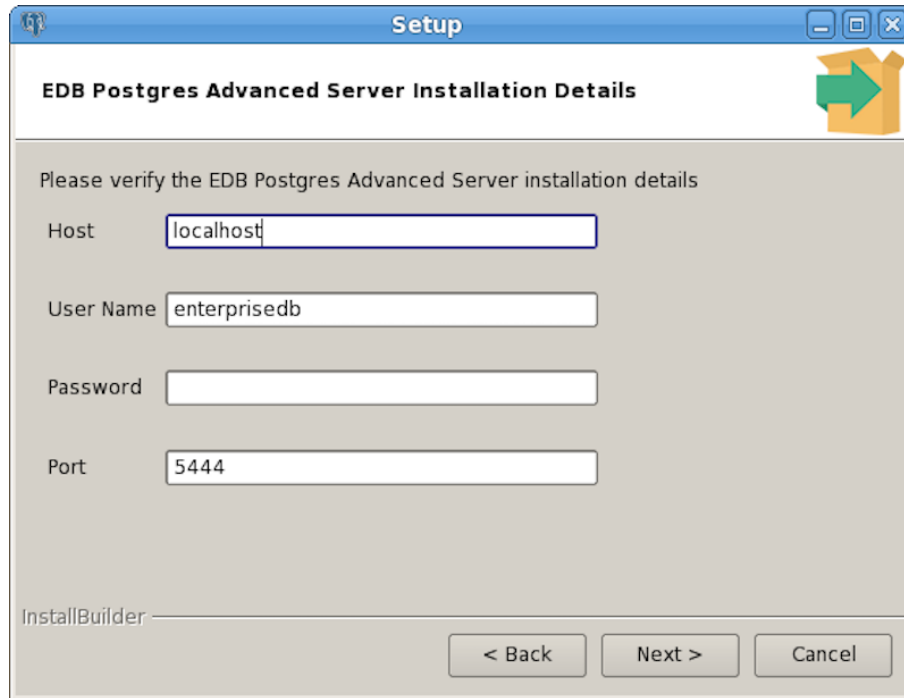
The image shows a Windows-style dialog box titled "Setup" with a standard minimize, maximize, and close button set in the top right corner. The main title bar of the dialog is "EDB Postgres Advanced Server Installation Details". Below the title bar, there is a green arrow pointing right, partially obscured by a yellow box. The main content area has a light gray background and contains the text "Please verify the EDB Postgres Advanced Server installation details". Below this text are four input fields: "Host" with the text "localhost", "User Name" with the text "enterprisedb", "Password" which is empty, and "Port" with the text "5444". At the bottom left of the dialog, the text "InstallBuilder" is visible. At the bottom right, there are three buttons: "< Back", "Next >", and "Cancel".

Fig. 6: *The Advanced Server installation details window*

9. The `Ready to Install` window notifies you when the installer has all of the information needed to install PostGIS on your system. Click `Next`.

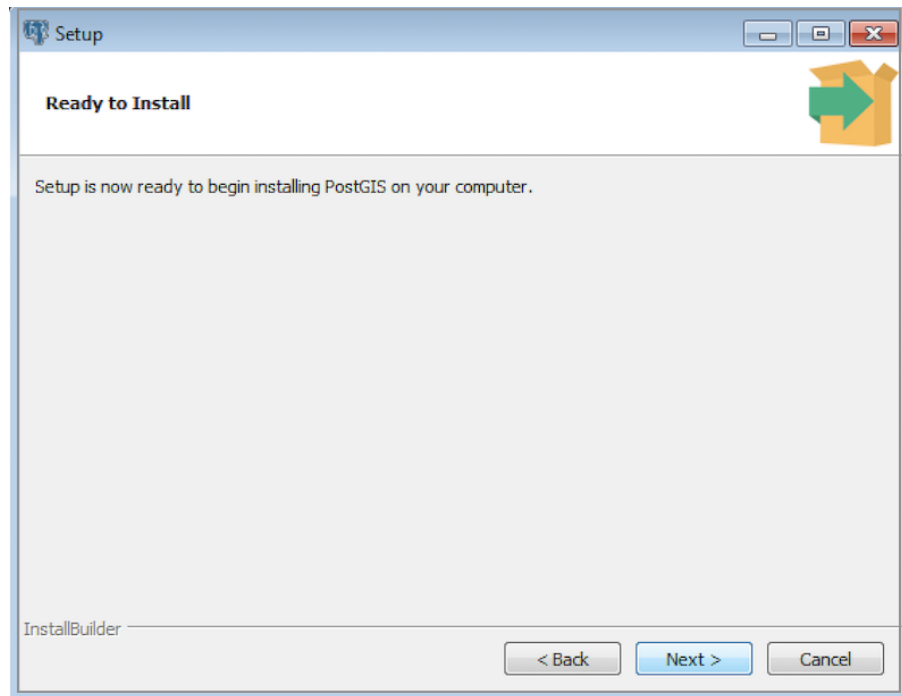


Fig. 7: The ready to install window

10. Progress bars inform you as the installation progresses; click **Finish** to exit the installer when the PostGIS installation completes.

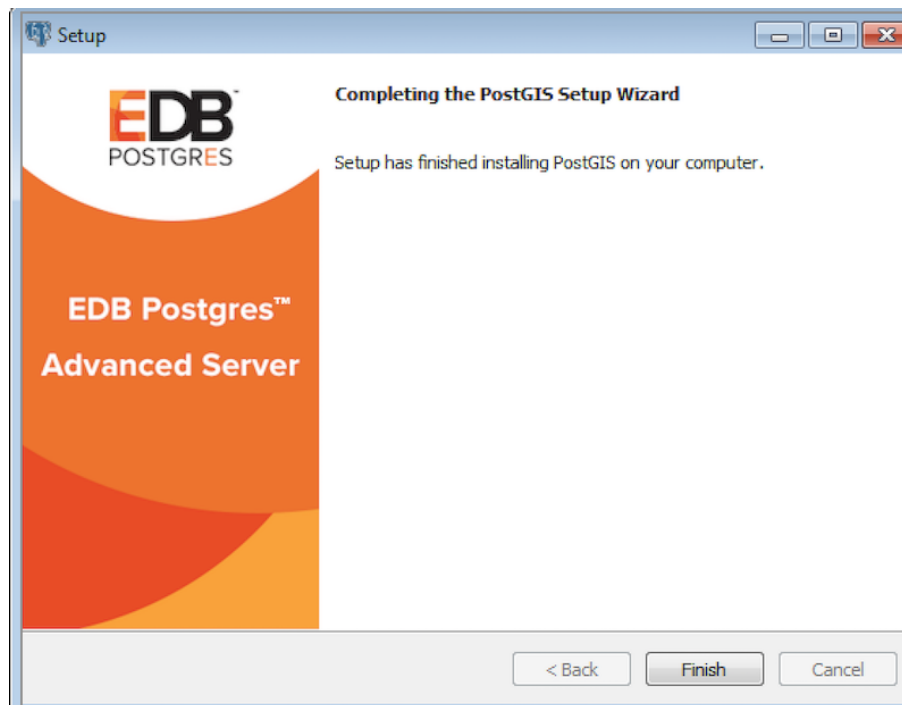


Fig. 8: The installation is complete

StackBuilder Plus will install PostGIS, and create the `template_postgis` database and PostGIS functions.

CHAPTER 2

Creating Extensions

After installing PostGIS, you can create a template database and create the PostGIS extensions. You must create the extensions in each database in which you will use PostGIS. You should not create the extensions in the `postgres` or `edb` database.

1. Before creating the `postgis` database, we recommend that you create a superuser to administer the database. To create the user, navigate into the `bin` directory under your Advanced Server installation and connect to the server with the `psql` client:

```
./psql -d edb -U enterprisedb -h 127.0.0.1
```

2. Then, to create a privileged role, invoke the following command:

```
CREATE ROLE gisadmin LOGIN PASSWORD 'password' SUPERUSER;
```

3. Log out of `psql`, then connect as `gisadmin`:

```
edb=# \q
./psql -d edb -U gisadmin -h 127.0.0.1
```

4. Use the following commands to create the `postgis` database owned by `gisadmin`:

```
CREATE DATABASE postgis;
```

5. Use the `\c` command to switch to the `postgis` database, and use the `CREATE EXTENSION` command to create the PostGIS Extensions:

```
\c postgis
CREATE EXTENSION postgis;
CREATE EXTENSION postgis_topology;
CREATE EXTENSION postgis_sfcgal;
CREATE EXTENSION fuzzystmatch;
CREATE EXTENSION address_standardizer;
CREATE EXTENSION address_standardizer_data_us;
CREATE EXTENSION postgis_tiger_geocoder;
```

When connected with pgAdmin, you should now see PostGIS extensions, functions, tables, and trigger functions beneath the `postgis` database public schema.

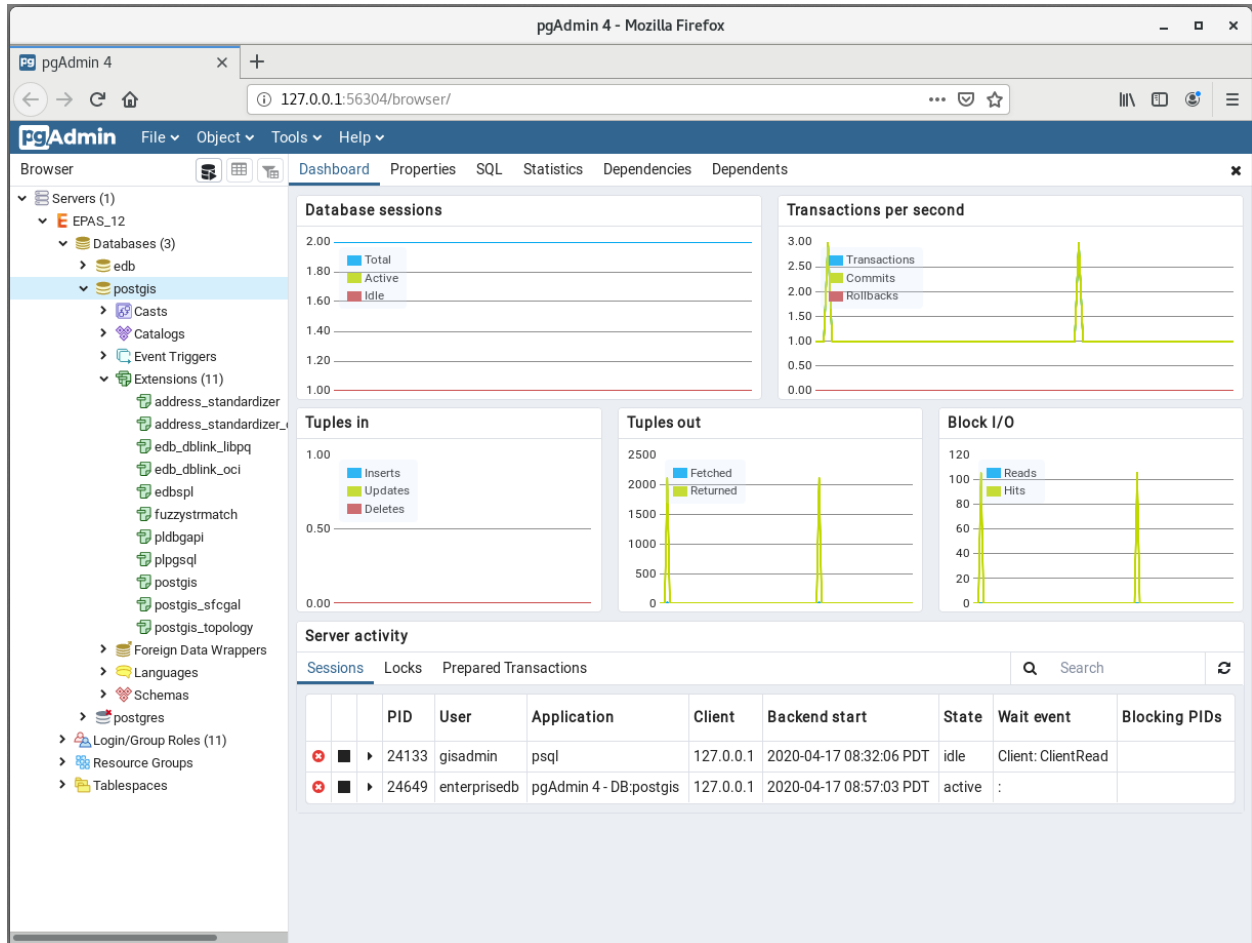


Fig. 1: PgAdmin displaying the PostGIS extensions

The `postgis` database is now geospatially enabled and can be used as a template to create new Geospatial databases.

CHAPTER 3

Using PostGIS

The following examples use PostGIS functions to create and query spatial objects. For more information about the functions that are part of PostGIS, please consult [the official PostGIS documentation](#).

The following command creates a table named `roads` table that will hold GIS data and a geometry column.

```
CREATE TABLE roads ( ID int4, NAME varchar(128) );
```

Then, use the PostGIS `AddGeometryColumn` function to add a column to the table:

```
SELECT AddGeometryColumn( 'roads', 'geom', -1, 'GEOMETRY', 2 );
```

Use the following SQL commands to insert data into the table `roads`. This data consists of the geometry of the type of `Linestring` (a line between 2 points):

```
INSERT INTO ROADS (ID,GEOM,NAME ) VALUES (1,ST_GeomFromText('LINESTRING(0 10,0 0)','-1),'Beacon Road');
INSERT INTO ROADS (ID,GEOM,NAME ) VALUES (2,ST_GeomFromText('LINESTRING(0 0,0 10)','-1),'Violet Road');
INSERT INTO ROADS (ID,GEOM,NAME ) VALUES (3,ST_GeomFromText('LINESTRING(0 0,10 0)','-1),'Skelton Street');
INSERT INTO ROADS (ID,GEOM,NAME ) VALUES (4,ST_GeomFromText('LINESTRING(0 0,10 10)','-1),'Fifth Avenue');
INSERT INTO ROADS (ID,GEOM,NAME ) VALUES (5,ST_GeomFromText('LINESTRING(0 10,0 0)','-1),'Main Street');
INSERT INTO ROADS (ID,GEOM,NAME ) VALUES (6,ST_GeomFromText('LINESTRING(10 0,0 0)','-1),'Lipton Street');
```

You can use the `GIST` function to create an index on the geometry column:

```
CREATE INDEX roads_index ON roads using GIST (geom);
```

`AsText(geometry)` is a PostGIS functions that returns a text representation of the geometry:

```
SELECT id, ST_AsText(geom) AS geom, name FROM ROADS order by id;
```

id	geom	name
1	LINESTRING(0 10,0 0)	Bacon Road
2	LINESTRING(0 0,0 10)	Violet Road
3	LINESTRING(0 0,10 0)	Skelton Street
4	LINESTRING(0 0,10 10)	Fifth Avenue
5	LINESTRING(0 10,0 0)	Main Street
6	LINESTRING(10 0,0 0)	Lipton Street

(6 rows)

After an index has been created, the && operator can be used in a query:

```
SELECT NAME, ST_AsText(GEOM) FROM ROADS WHERE GEOM && SetSRID('BOX3D(10 10,10 10)
'::box3d,-1);
```

name	astext
Fifth Avenue	LINESTRING(0 0,10 10)

(1 row)

The BOX3D function is used to specify a bounding box. The && operator uses the index to quickly reduce the result set down to only those geometries which have bounding boxes that overlap the specified area.

You can use the ~= operator to check if two geometries are geometrically identical:

```
SELECT ID, NAME FROM roads WHERE GEOM ~= ST_GeomFromText('LINESTRING(0 10,0 0)',-1)
order by id;
```

id	name
1	Bacon Road
5	Main Street

(2 rows)

Upgrading PostGIS

This section outlines the process of upgrading PostGIS. Perform the following steps to upgrade from older versions of PostGIS to the latest version.

1. Assume the identity of the root user and invoke the following command to upgrade to PostGIS 3.0.1:

```
yum upgrade edb-as12-postgis-3.0.1
```

2. Connect to the database (where you have already created the extensions) with the psql client application, and execute the following commands to update the extensions:

```
edb# SELECT PostGIS_Extensions_Upgrade();
edb=# ALTER EXTENSION postgis UPDATE TO '3.0.1';
edb=# ALTER EXTENSION postgis_topology UPDATE TO '3.0.1';
edb=# ALTER EXTENSION postgis_tiger_geocoder UPDATE TO '3.0.1';
edb=# ALTER EXTENSION address_standardizer UPDATE TO '3.0.1';
edb=# ALTER EXTENSION postgis_sfcgal UPDATE TO '3.0.1';
```

3. If the `postgis_raster` extension exists, you must remove it with the following command:

```
DROP EXTENSION postgis_raster;
```

Uninstalling PostGIS

Uninstalling an RPM

To uninstall PostGIS, assume the identity of the root user and invoke the following command:

```
yum erase edb-as<xx>-postgis-*<y.y.y>
```

Where <xx> is the Advanced Server version and <y.y.y> is the PostGIS version you want to uninstall.

Using a Graphical Uninstaller to remove PostGIS

The PostGIS graphical installer creates an uninstaller that you can use to remove PostGIS. The uninstaller is created in the installation directory that you have specified while installing PostGIS (default is C:\Program Files\edb\as12).

1. Navigate into the directory that contains the uninstaller and assume superuser privileges. Open the uninstaller and click **Yes** to begin uninstalling PostGIS:

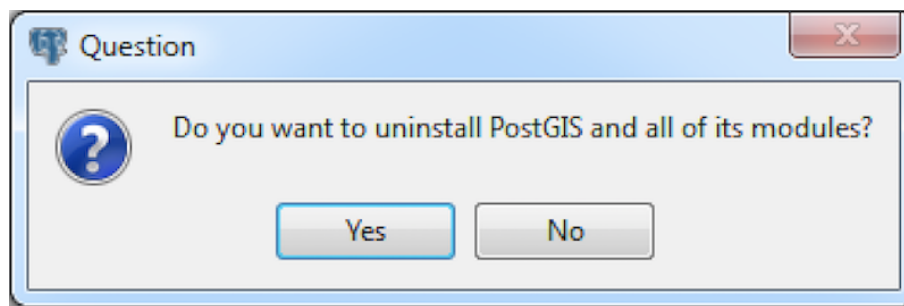


Fig. 1: *The Uninstaller opens*

2. The uninstallation process begins. Click **OK** when the uninstallation completes:

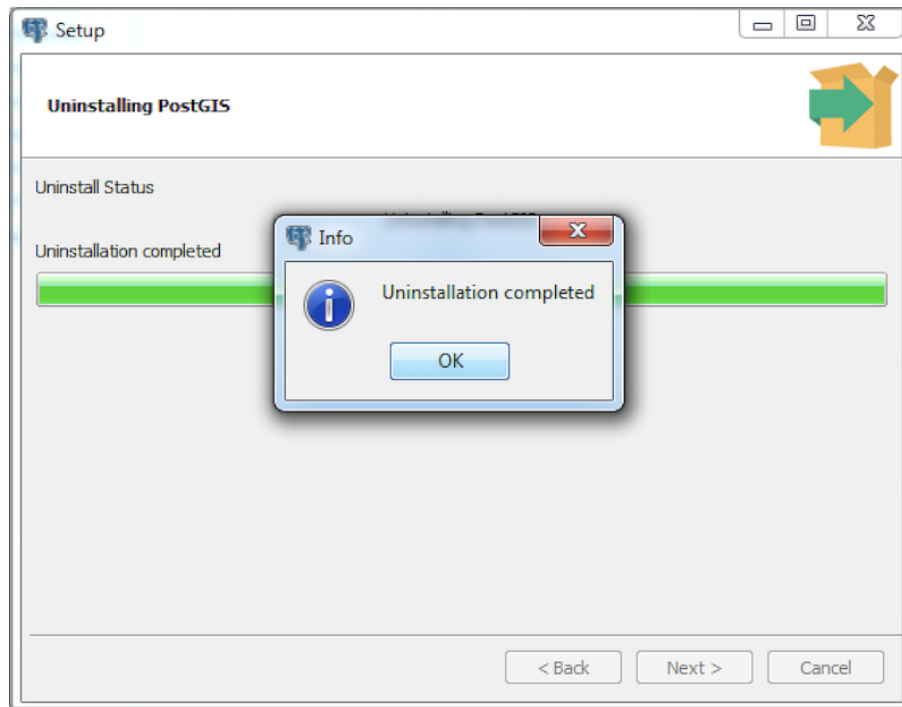


Fig. 2: *Uninstallation is Complete*

CHAPTER 6

Conclusion

EDB Postgres PostGIS User's Guide

Copyright © 2020 EnterpriseDB Corporation.

All rights reserved.

EnterpriseDB® Corporation

34 Crosby Drive, Suite 201, Bedford, MA 01730, USA

T +1 781 357 3390 F +1 978 467 1307 E

info@enterprisedb.com

www.enterprisedb.com

- EnterpriseDB and Postgres Enterprise Manager are registered trademarks of EnterpriseDB Corporation. EDB and EDB Postgres are trademarks of EnterpriseDB Corporation. Oracle is a registered trademark of Oracle, Inc. Other trademarks may be trademarks of their respective owners.
- EDB designs, establishes coding best practices, reviews, and verifies input validation for the logon UI for EDB Postgres product where present. EDB follows the same approach for additional input components, however the nature of the product may require that it accepts freeform SQL, WMI or other strings to be entered and submitted by trusted users for which limited validation is possible. In such cases it is not possible to prevent users from entering incorrect or otherwise dangerous inputs.
- EDB reserves the right to add features to products that accept freeform SQL, WMI or other potentially dangerous inputs from authenticated, trusted users in the future, but will ensure all such features are designed and tested to ensure they provide the minimum possible risk, and where possible, require superuser or equivalent privileges.
- EDB does not warrant that we can or will anticipate all potential threats and therefore our process cannot fully guarantee that all potential vulnerabilities have been addressed or considered.

C

Conclusion, [16](#)

Creating Extensions, [9](#)

I

Installing PostGIS, [2](#)

U

Uninstalling PostGIS, [14](#)

Upgrading PostGIS, [13](#)

Using PostGIS, [11](#)

Using StackBuilder Plus to Install
PostGIS, [4](#)