Contents

---

0 PostGIS

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.

---

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.

Installing PostGIS on a CentOS/RHEL Host

You must install Advanced Server before installing PostGIS. For details about installing and configuring Advanced Server, see the EDB Postgres Advanced Server Installation Guide.

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

Before creating the repository configuration file, you must have credentials that allow access to the EnterpriseDB repository. For information about requesting credentials, visit this page.

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

On CentOS/RHEL 7:

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

On CentOS/RHEL 8:

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

The repository configuration file is named `edb.repo` , which resides in `/etc/yum.repos.d` .

2. Use your choice of editor to open the file. Ensure that the value of the enabled parameter is `1` and the `<username>` and `<password>` placeholders in the `baseurl` specification are replaced with the username 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
   ```

3. Save the configuration file and exit the editor.

4. Before installing PostGIS on a CentOS/RHEL 8 machine, you need to enable the `PowerTools` repo with the following command:

   ```
   dnf config-manager --set-enabled PowerTools
   ```

5. Install `PostGIS` with one of the following platform-specific command:

On CentOS/RHEL 7:

```
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
```

On CentOS/RHEL 8:

```
dnf 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:

```
dnf 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.

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

Installing PostGIS on a Debian/Ubuntu Host

You must install Advanced Server before installing PostGIS. For details about installing and configuring Advanced Server, see the EDB Postgres Advanced Server Installation Guide.

To install a package on a Debian or Ubuntu host, you must have credentials to access the EnterpriseDB repository. If you need EnterpriseDB credentials, visit the EnterpriseDB website.

The following steps will walk you through using the EnterpriseDB repository to install a DEB package. When using the commands, replace the `username` and `password` with the credentials provided by EnterpriseDB.

Note

If you are using the pdf version of this document, using cut/paste to copy command may result in extra spaces or carriage returns in the pasted command. If a command fails, check the command carefully for additional characters.

1. Assume superuser privileges:

   ```
   sudo su -
   ```

2. Configure the EnterpriseDB repository:

```
sh -c 'echo "deb https://<username>:<password>@apt.enterprisedb.com/$(lsb_release -cs)-edb $(lsb_release -cs) main" > /etc/apt/sources.list.d/edb-$(lsb_release -cs).list'
```

3. Add support to your system for secure APT repositories:

```
apt-get install apt-transport-https
```

4. Add the EDB signing key:

```
wget -q -O - https://<username>:<password>@apt.enterprisedb.com/edb-deb.gpg.key | apt-key add -
```

5. Update the repository metadata:

```
apt-get update
```

6. Install DEB package:

```
apt-get install edb-as<xx>-postgis-<y.y>
```

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

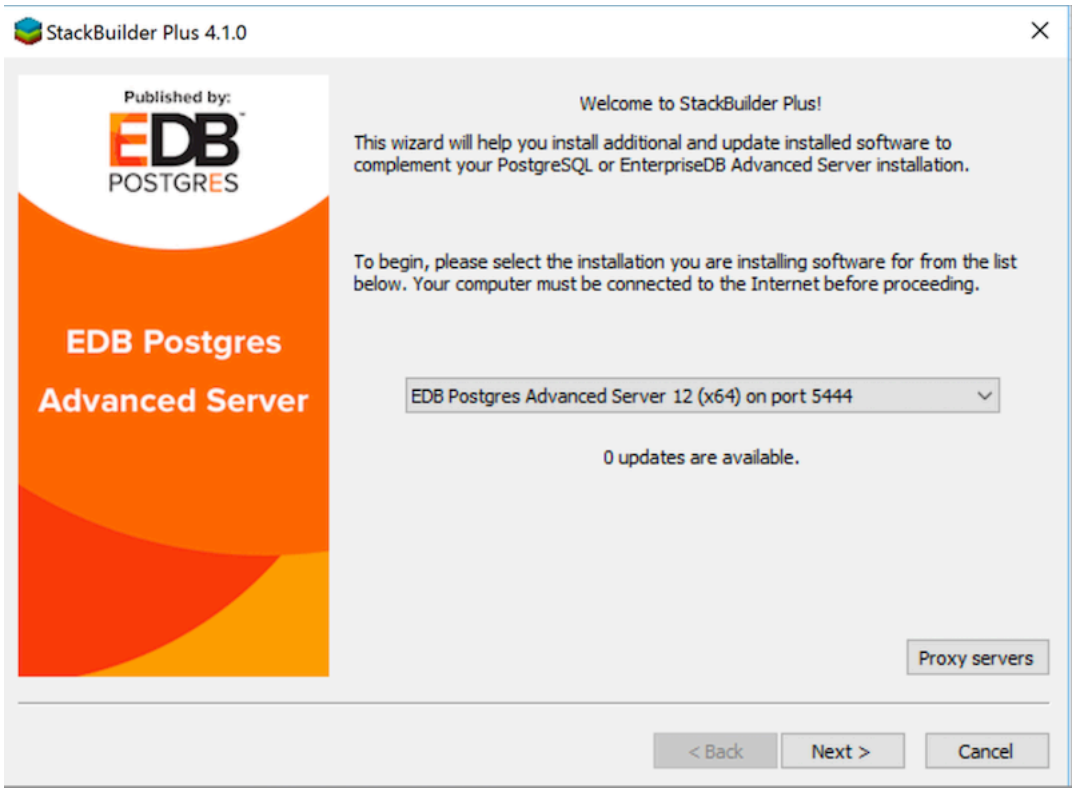
For example, to install the PostGIS 3.0.1 package for Advanced Server 12, execute the following command:

```
apt-get install edb-as12-postgis-3.0
```

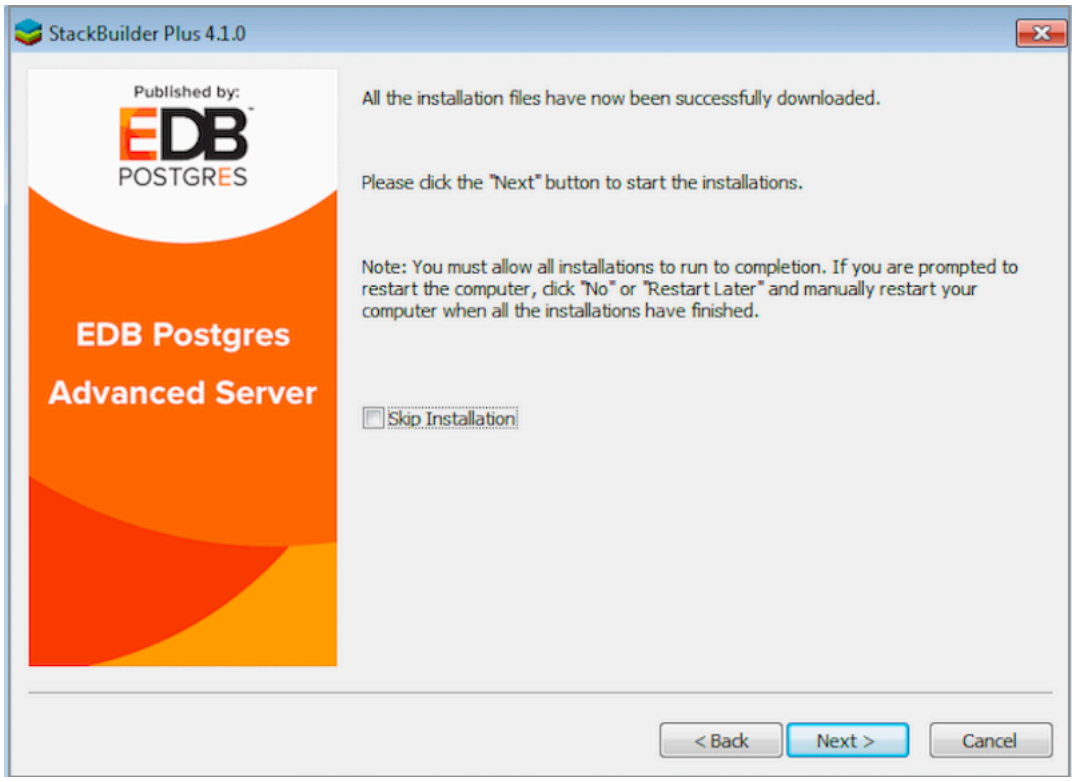
Installing PostGIS on a Windows Host

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.

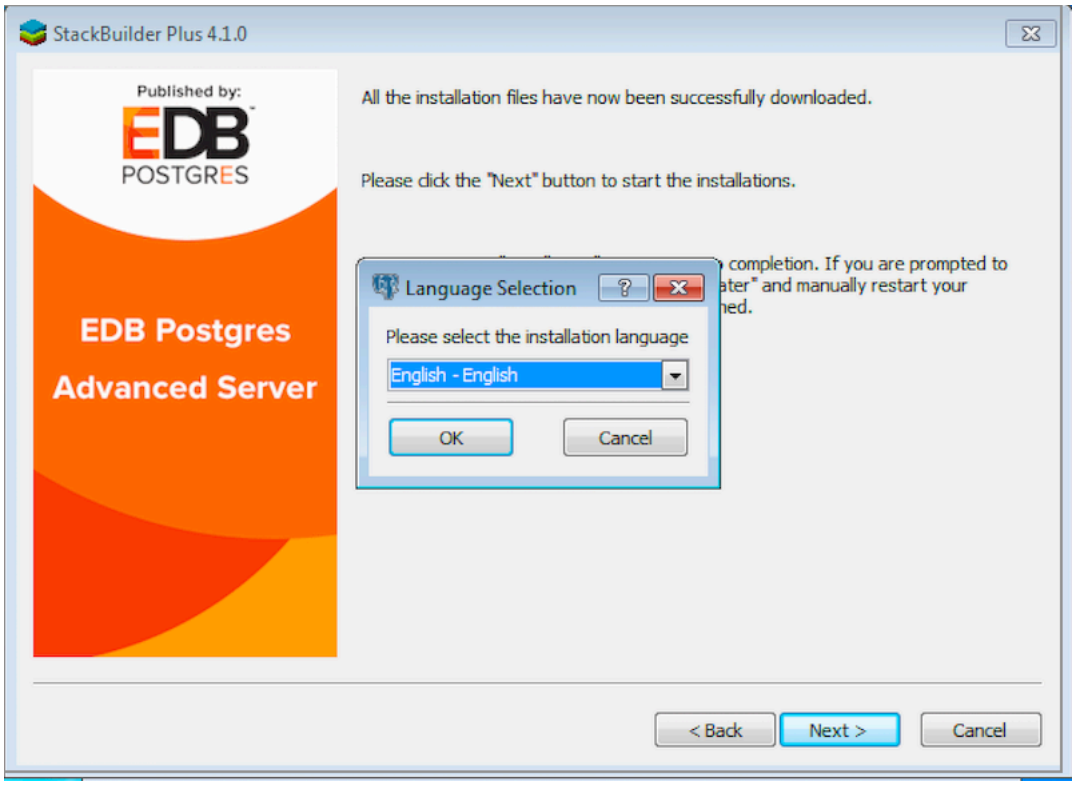
1. Open StackBuilder Plus and select your Advanced Server installation from the drop-down list on the `Welcome` window. Click `Next` to continue to the application selection page.



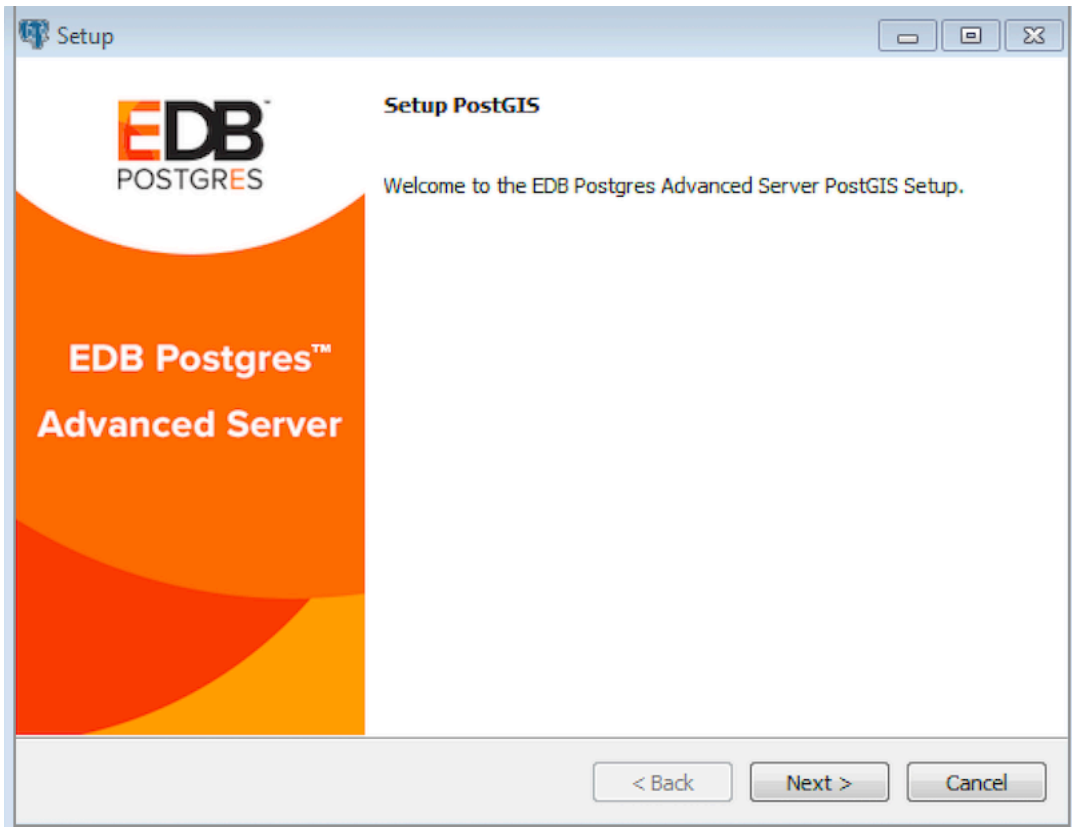
2. Expand the `Spatial Extensions` node, and check the box next to the PostGIS version. Click `Next` to continue.
3. The selected packages and the default download directory where the package will be installed are displayed; change the locations if required. Click `Next` .
4. Once you have downloaded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



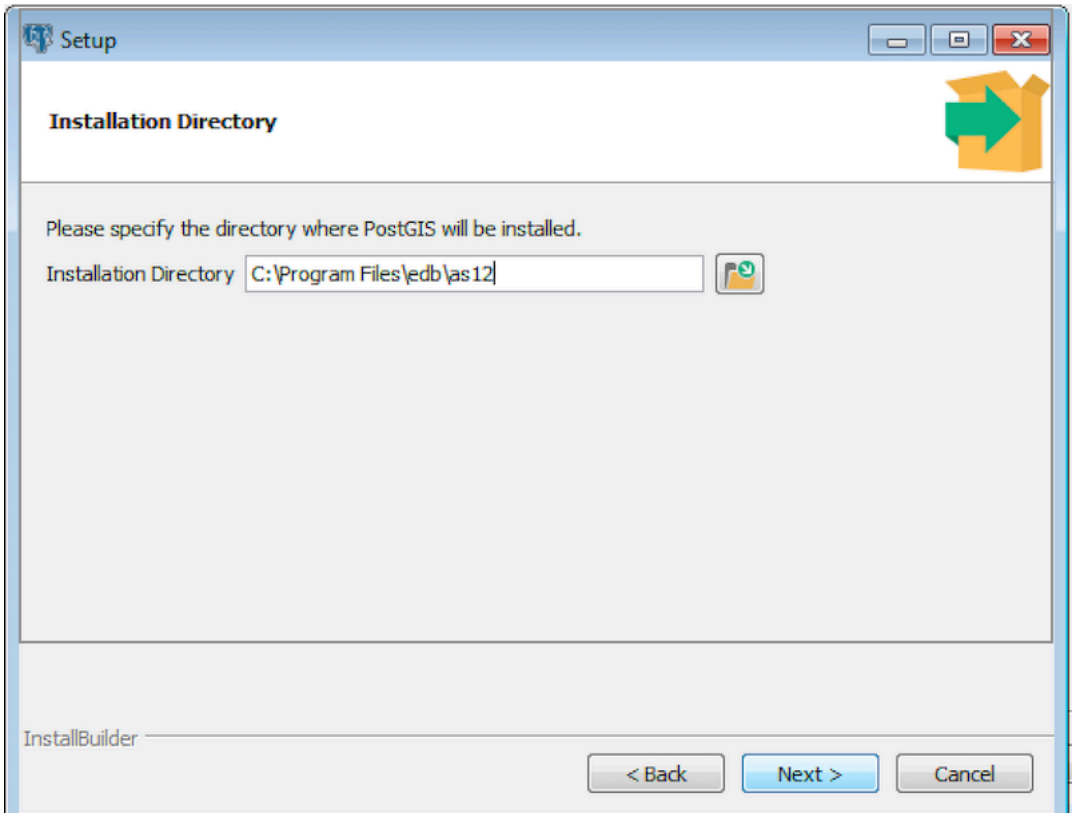
5. Select an installation language and click `OK` .



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



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

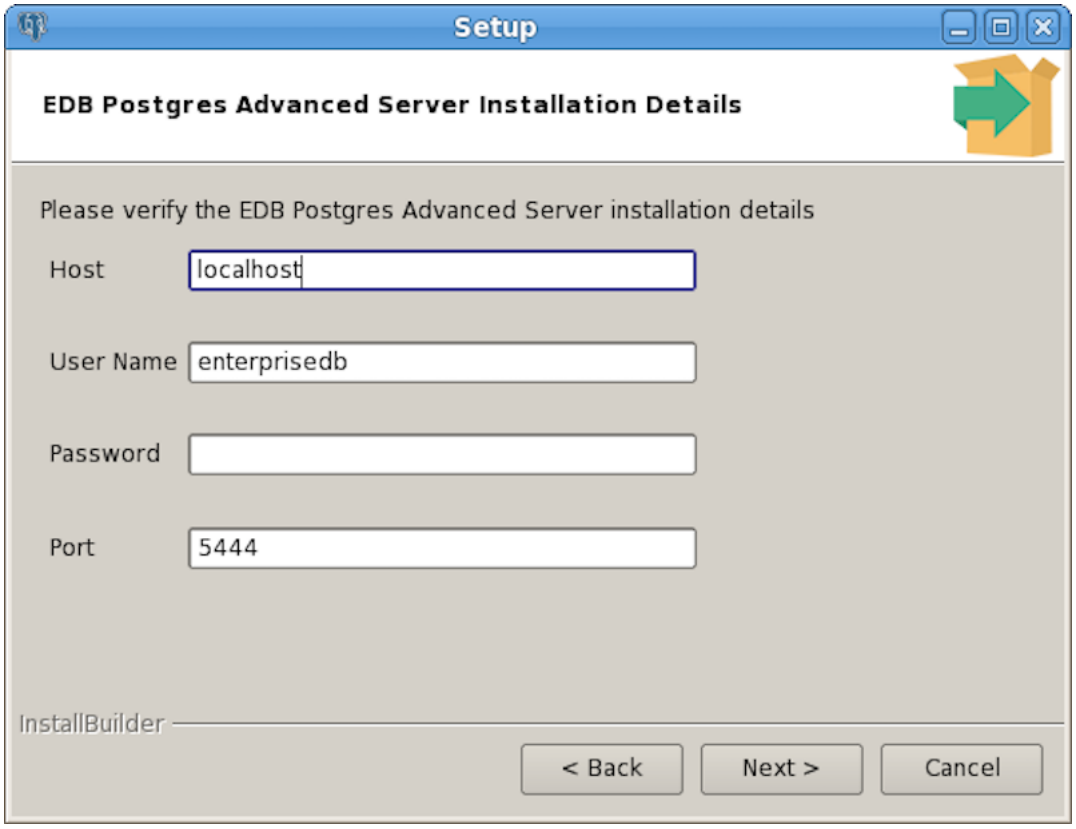


8. Use fields on the `EDB Postgres Advanced Server Installation Details` window to provide connection information for the Advanced Server host:

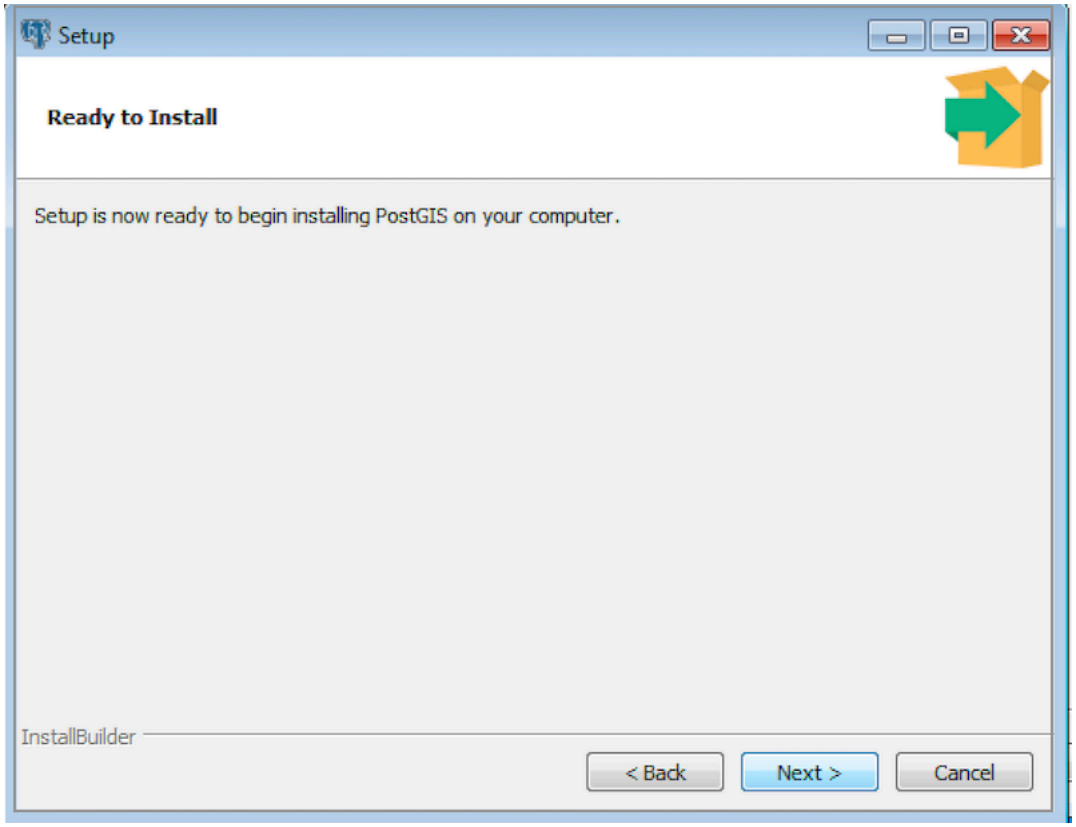
- Use the `Host` field to identify the system on which Advanced Server resides.
- Provide the name of the role that PostGIS will use for connections to the server in the `User Name` field.

- 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.

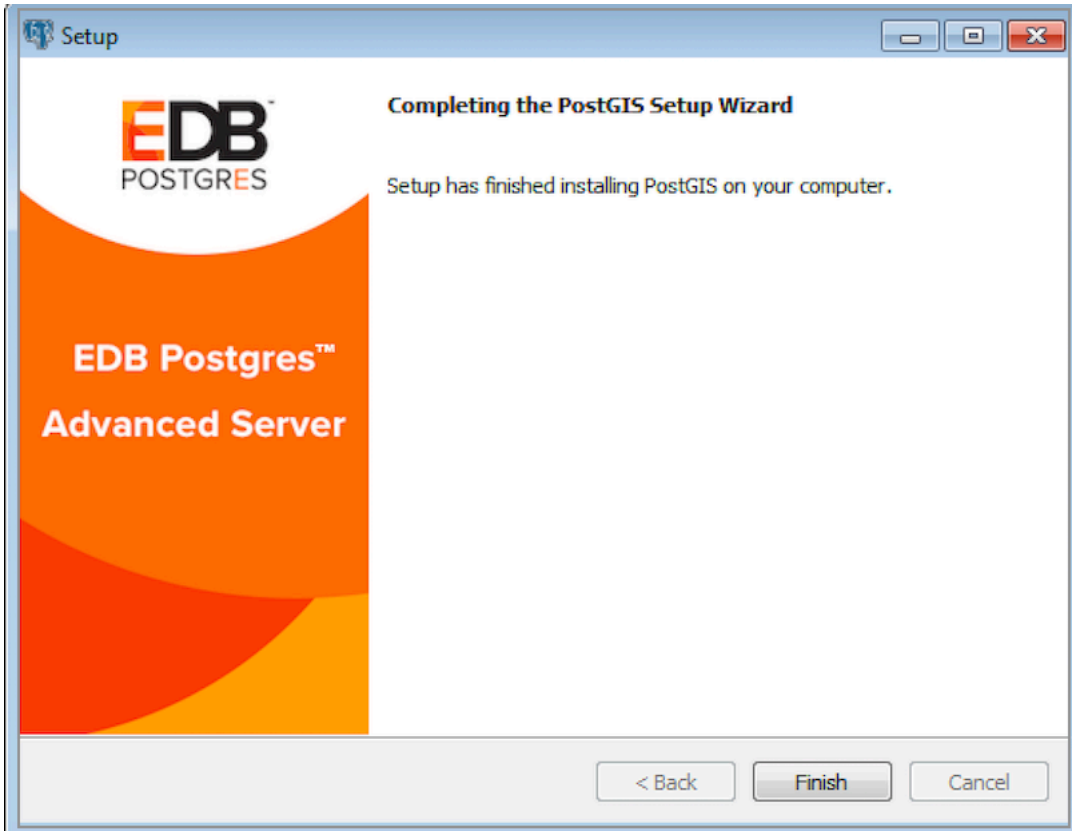
Then, click `Next` to continue.



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` .



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



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

---

2 Creating Extensions

After installing PostGIS, create a PostGIS database and the extensions in each database you wish to use PostGIS extensions. You must not create the extensions in the `postgres` or `edb` database.

1. Before creating the `postgis` database, we recommend creating 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` :

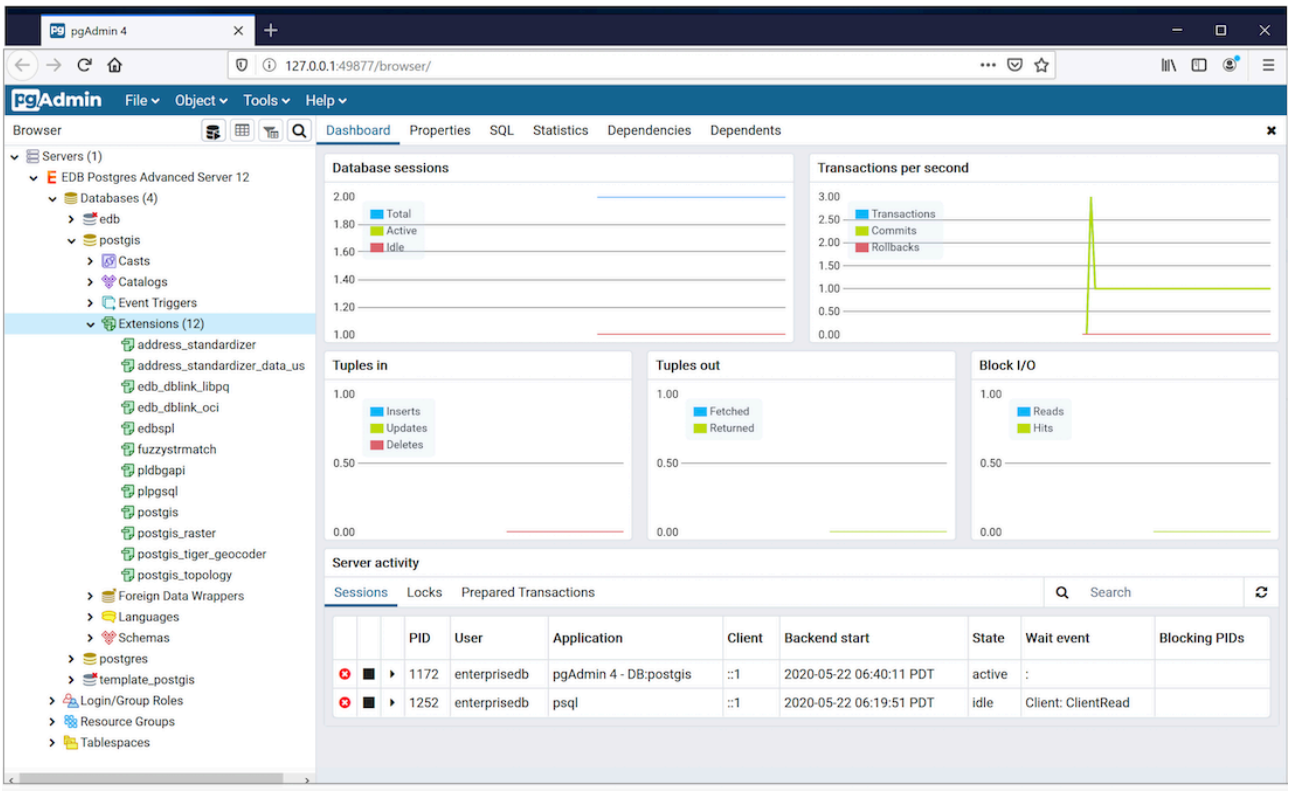
```
CREATE DATABASE postgis;
```

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 fuzzystrmatch;
```

```
CREATE EXTENSION address_standardizer;
CREATE EXTENSION address_standardizer_data_us;
CREATE EXTENSION postgis_tiger_geocoder;
CREATE EXTENSION postgis_sfcgal;
CREATE EXTENSION postgis_raster;
```

When connected with pgAdmin, you should now see PostGIS extensions, functions, tables, and trigger functions beneath the `postgis` database public schema. The `postgis` database is now geospatially enabled and can be used as a template to create new Geospatial databases.



3 Upgrading PostGIS

This section outlines the process of upgrading PostGIS.

To upgrade between major versions, for example, to upgrade from 2.5.4 to 3.0.1, perform the following steps:

1. Install PostGIS 3.0.1.
2. To update extensions, connect to the database (where you have already added extensions) with the psql client application, and execute the following commands:

```
edb=# alter extension postgis update TO “3.0.1”;
WARNING:  unpackaging raster
WARNING:  PostGIS Raster functionality has been unpackaged
HINT:  type `SELECT postgis_extensions_upgrade(); to finish the upgrade. After upgrading, if y
ALTER EXTENSION
edb=# SELECT postgis_extensions_upgrade();
NOTICE:  Packaging extension postgis_raster
NOTICE:  Updating extension postgis_topology from 2.5.4 to 3.0.1
NOTICE:  Updating extension postgis_tiger_geocoder from 2.5.4 to 3.0.1
                    postgis_extensions_upgrade
--------------------------------------------------------------------
 Upgrade completed, run SELECT postgis_full_version(); for details
(1 row)
```

4 Using PostGIS

The following examples use PostGIS functions to create and query spatial objects. For more information about the PostGIS functions, 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)
```

---

5 Uninstalling PostGIS

Uninstalling PostGIS on a CentOS/RHEL Host

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

On CentOS/RHEL7:

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

On CentOS/RHEL8:

```
dnf 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.

Uninstalling PostGIS on a Debian/Ubuntu Host

To uninstall PostGIS on a Debian or Ubuntu platform, invoke the following command:

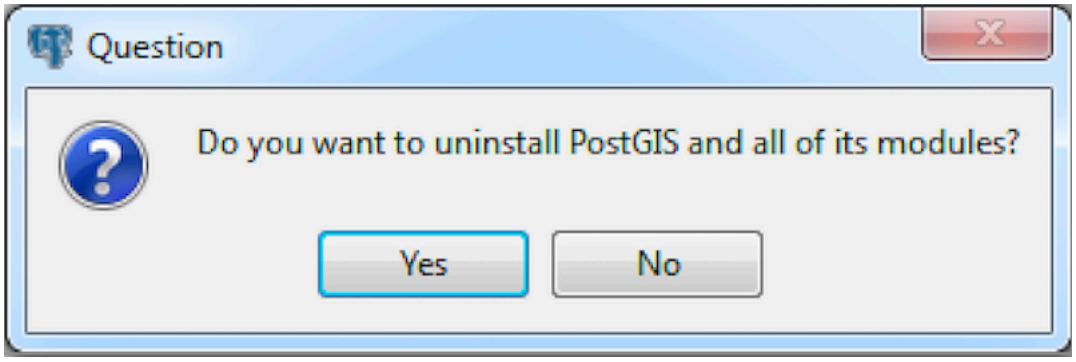
```
apt-get remove edb-as<xx>-postgis-<y.y>*
```

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

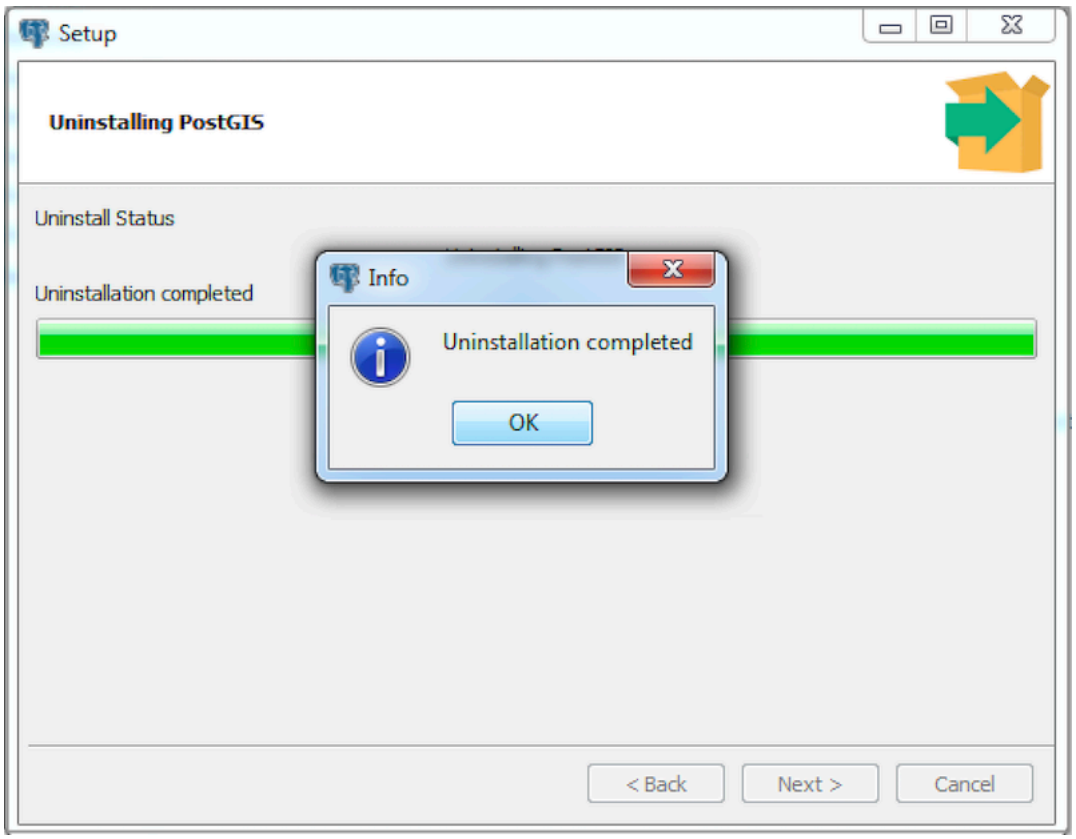
Uninstalling PostGIS on a Windows Host

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:



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


Setup
Uninstalling PostGIS
Uninstall Status
Uninstallation completed
Info
Uninstallation completed
OK
< Back
Next >
Cancel