PostGIS

Version 3.1.2

Built at 2021-07-28T12:02:23

1 PostGIS

EDB PostGIS is a PostgreSQL extension that allows you to store Geographic Information Systems (GIS) objects in an Advanced Server database. It 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.

What's New

This release contains a merge with upstream, which includes the following bug fixes:

- Improved handling of more complex compound coordinate systems.
- The SRID in TopoGeometry is fixed for empty geometry objects.
- Improved message when no zip code is provided for `postgis_tiger_geocoder`.
- Axis flips are now done for a CRS when `Lat` is set as the first column. In other cases, EPSG database order is used.
- Support recent Proj versions that have removed `pj_get_release`.
- The tolerance is now adjusted to improve geodetic calculations.
- Fix on improper conversion of negative geographic azimuth to positive.
- Allow DBSCAN cluster to form when recordset length equal to minPoints.
- Ensure bounding boxes are refreshed after coordinate transforms.
- Fix raster issues related to PostgreSQL 14 tablefunc changes.
- Update to support Tiger 2020.
- Change Proj cache lifetime to last as long as connection.

For more information, please refer to the Upstream release notes.

!!! Note To upgrade from PostGIS version 2.5.4 (or lower) to 3.1.2, you must perform the following actions: - Since the return type of the raster functions have changed, you must drop and re-create the raster extension as part of the upgrade process. - Before upgrading to version 3.1.2, you must upgrade to 3.1.1 first as an intermediate step. - When the PostGIS data has a dependency on the raster functions, upgrading to PostGIS 3.1.2 requires dumping and reloading the data.

```
See the [upgrade section]
(https://www.enterprisedb.com/docs/postgis/latest/03_upgrading_postgis/) for
details.
```

2 Installing PostGIS

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

!!! Note 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	Supported Platforms
PostGIS 3.1	Advanced Server 13	RHEL/CentOS 7 and RHEL/CentOS 8 - x86_64 RHEL/CentOS 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver and Ubuntu 20.04 LTS Focal Fossa Windows 64 x86 Interactive Installer
PostGIS 3.1	Advanced Server 12	RHEL/CentOS 7 x86_64 and RHEL/CentOS 8 - x86_64 RHEL/CentOS 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver Windows 64 x86 Interactive Installer SLES 12
PostGIS 3.1	Advanced Server 11	RHEL/CentOS 7 - x86_64 and RHEL/CentOS 8 - x86_64 RHEL/CentOS 7 - ppc64le Debian 9x Stretch Ubuntu 18.04 LTS Bionic Beaver Windows 64 x86 Interactive Installer
PostGIS 3.1	Advanced Server 9.6, 10	RHEL/CentOS 7 - x86_64 RHEL/CentOS 7 - ppc64le Windows 64 x86 Interactive Installer
PostGIS 3.0	Advanced Server 13	RHEL/CentOS 7 and RHEL/CentOS 8 - x86_64 RHEL/CentOS 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver and Ubuntu 20.04 LTS Focal Fossa Windows 64 x86 Interactive Installer
PostGIS 3.0	Advanced Server 12	RHEL/CentOS 7 x86_64 and RHEL/CentOS 8 - x86_64 RHEL/CentOS 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver Windows 64 x86 Interactive Installer
PostGIS 3.0	Advanced Server 11	RHEL/CentOS 7 - x86_64
PostGIS 2.5	Advanced Server 12	RHEL/CentOS 7 - x86_64 and RHEL/CentOS 8 - x86_64 RHEL/CentOS 7 - ppc64le Debian 9x Stretch and 10x Buster Ubuntu 18.04 LTS Bionic Beaver Windows 64 x86 Interactive Installer SLES 12
PostGIS 2.5	Advanced Server 11	RHEL/CentOS 7 - x86_64 RHEL/CentOS 7 - ppc64le Debian 9x Stretch Ubuntu 18.04 LTS Bionic Beaver Windows 64 x86 Interactive Installer SLES 12
PostGIS 2.4	Advanced Server 9.6, 10 and 11	RHEL/CentOS 7 - x86_64 RHEL/CentOS 7 - ppc64le Windows 64 x86 Interactive Installer

PostGIS Version	Supported Advanced Server Versions	Supported Platforms
`PostGIS 2.3`	Advanced Server 9.6 and 10	RHEL/CentOS 7 - x86_64 RHEL/CentOS 7 - ppc64le Windows 64 x86 Interactive Installer

!!! Note PostGIS is no longer supported on CentOS/RHEL/OL 6.x platforms. It is strongly recommended that EDB products running on these platforms be migrated to a supported platform.

Installing PostGIS on a CentOS Host

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

Perform the following steps to install PostGIS on a CentOS Host:

1. To create the repository configuration file, assume superuser privileges, and invoke the platform-specific command:

   On CentOS 7:

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

   On CentOS 8:

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

2. Replace the `USERNAME:PASSWORD` variable in the following command with the username and password of a registered EnterpriseDB user:

   ```
   sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
   ```

3. Before installing PostGIS, you need to install the Extra Packages for Enterprise Linux (EPEL) release package:

   On CentOS 7:

   ```
   yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
   ```

   On CentOS 8:

   ```
   dnf -y install epel-release
   ```

4. The following steps are applicable only for CentOS 8 platform:

   a. Enable the PowerTools repository to satisfy package dependencies:

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

   b. Disable the built-in PostgreSQL module:

   ```
   dnf -qy module disable postgresql
   ```

5. Invoke the platform-specific command to install `PostGIS`:

On CentOS 7, to install the latest version of PostGIS version for EDB Advanced Server version 13.0:

```
yum -y install edb-as13-postgis3
```

On CentOS 8, to install the latest version of PostGIS version for EDB Advanced Server version 13.0:

```
dnf -y install edb-as13-postgis3
```

On CentOS 7, to install the latest version of PostGIS for EDB Advanced Server versions older than 13.0:

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

Where *<xx>* denotes the Advanced Server version older than 13.0.

On CentOS 7, to install PostGIS older versions (for example PostGIS 3.0.2) for EDB Advanced Server version 13.0:

```
yum -y install edb-as13-postgis3-3.0.2
```

On CentOS 7, to install older PostGIS versions for EDB Advanced Server versions older than 13.0:

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

Where *<xx>* is the Advanced Server version older than 13.0 and*<y.y.y>* is the older PostGIS version.

For example, to install PostGIS 2.5.4 on Advanced Server 12:

```
yum -y install edb-as12-postgis-2.5.4
```

When you install an RPM package 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.

!!! Note After installing PostGIS with a package manager, you must manually create a template database and the required PostGIS extension.

Installing PostGIS on an RHEL Host

Before installing the repository configuration file, you must have credentials that allow access to the EnterpriseDB repository. For information about requesting credentials, visit the EDB website.

Perform the following steps to install PostGIS on an RHEL Host:

1. To create the repository configuration file, assume superuser privileges, and invoke the platform-specific command:

   On RHEL 7:

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

   On RHEL 8:

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

2. Replace the `USERNAME:PASSWORD` variable with the username and password of a registered EnterpriseDB user:

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

3. Before installing PostGIS, you need to install the Extra Packages for Enterprise Linux (EPEL) release package:

   On RHEL 7:

   ```
   yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
   ```

   On RHEL 8:

   ```
   dnf -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
   ```

4. Enable additional platform-specific repositories:

   On RHEL 7, enable the `optional, extras`, and `HA` repositories to satisfy package dependencies:

   ```
   subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-extras-rpms"  --enable "rhel-ha-for-rhel-*-server-rpms"
   ```

   On RHEL 8, enable the `codeready-builder-for-rhel-8-*-rpms` repository to satisfy EPEL packages dependency:

   ```
   ARCH=$( /bin/arch )

   subscription-manager repos --enable "codeready-builder-for-rhel-8-${ARCH}-rpms"
   ```

5. On RHEL 8 platform, disable the built-in PostgreSQL module:

   ```
   dnf -qy module disable postgresql
   ```

6. Invoke the platform-specific command to install `PostGIS`:

   On RHEL 7, to install the latest version of PostGIS for EDB Advanced Server version 13.0:

   ```
   yum -y install edb-as13-postgis3
   ```

   On RHEL 8, to install the latest version of PostGIS for EDB Advanced Server version 13.0:

   ```
   dnf -y install edb-as13-postgis3
   ```

   On RHEL 7, to install the latest version of PostGIS for EDB Advanced Server version older than 13.0:

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

   Where *<xx>* denotes the Advanced Server version older than 13.0

   On RHEL 7, to install PostGIS older versions (for example PostGIS 3.0.2) for EDB Advanced Server version 13.0:

   ```
   yum -y install edb-as13-postgis3-3.0.2
   ```

   On RHEL 7, to install older PostGIS versions for EDB Advanced Server version older than 13.0:

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

Where *<xx>* is the Advanced Server version older than 13.0 and*<y.y.y>* is the older PostGIS version.

For example, to install PostGIS 2.5.4 on EDB Advanced Server version 12:

```
yum -y install edb-as12-postgis-2.5.4
```

When you install an RPM package 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.

!!! Note After installing PostGIS with a package manager, you must manually create a template database and the required PostGIS extension.

Installing PostGIS on an RHEL/CentOS 7 PPCLE Host

Before installing the repository configuration, you must have credentials that allow access to the EnterpriseDB repository. For information about requesting credentials, visit the EDB website.

Perform the following steps to install PostGIS on an RHEL/CentOS 7 PPC64LE Host.

1. Create the Advance Toolchain repository configuration file:

```
rpm --import
https://public.dhe.ibm.com/software/server/POWER/Linux/toolchain/at/redhat/RHEL7/gp
-pubkey-6976a827-5164221b

cat > /etc/yum.repos.d/advance-toolchain.repo <<EOF

# Beginning of the configuration file
[advance-toolchain]
name=Advance Toolchain IBM FTP
baseurl=https://public.dhe.ibm.com/software/server/POWER/Linux/toolchain/at/redhat/
7
failovermethod=priority
enabled=1
gpgcheck=1
gpgkey=ftp://public.dhe.ibm.com/software/server/POWER/Linux/toolchain/at/redhat/RHI
-pubkey-6976a827-5164221b
# End of the configuration file
```

2. To create the EnterpriseDB repository configuration file, assume superuser privileges and invoke the following command:

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

3. Replace the `USERNAME:PASSWORD` variable with the username and password of a registered EnterpriseDB user:

```
sed -i "s@<username>:<password>@USERNAME:PASSWORD@" /etc/yum.repos.d/edb.repo
```

4. Before installing PostGIS, you need to install the Extra Packages for Enterprise Linux (EPEL) release package:

```
yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-
7.noarch.rpm
```

5. On RHEL 7, enable the `optional, extras`, and `HA` repositories to satisfy package dependencies:

```
subscription-manager repos --enable "rhel-*-optional-rpms" --enable "rhel-*-
extras-rpms"  --enable "rhel-ha-for-rhel-*-server-rpms"
```

6. Invoke the following command to install PostGIS:

   To install the latest version of PostGIS version for EDB Advanced Server version 13.0:

```
yum -y install edb-as13-postgis3
```

   To install the latest version of PostGIS for EDB Advanced Server version older than 13.0:

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

   Where *<xx>* denotes the Advanced Server version older than 13.0

   To install PostGIS older versions (for example PostGIS 3.0.2) for EDB Advanced Server version 13.0:

```
yum -y install edb-as13-postgis3-3.0.2
```

   To install older PostGIS versions for EDB Advanced Server versions older than 13.0:

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

   Where *<xx>* is the Advanced Server version older than 13.0 and*<y.y.y>* is the older PostGIS version.

   For example, to install PostGIS 2.5.4 on EDB Advanced Server version 12:

```
yum -y install edb-as12-postgis-2.5.4
```

   When you install an RPM package 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.

   !!! Note After installing PostGIS with a package manager, you must manually create a template database and the required PostGIS extension.

Installing PostGIS on a Debian/Ubuntu Host

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

Perform the following steps to install PostGIS Debian package using the EnterpriseDB repository:

1. Assume superuser privileges:

```
sudo su -
```

2. Configure the EnterpriseDB repository.

   On Debian 9, Ubuntu 18, and Ubuntu 20, invoke the folloiwng command and substitute your EnterpriseDB credentials for the `username` and `password` placeholders.

```
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'
```

On Debian 10:

a. Set up the EnterpriseDB repository:

```
sh -c 'echo "deb [arch=amd64] https://apt.enterprisedb.com/$(lsb_release -cs)-
edb/ $(lsb_release -cs) main" > /etc/apt/sources.list.d/edb-$(lsb_release -
cs).list'
```

b. Substitute your EnterpriseDB credentials for the `username` and `password` placeholders in the following command:

```
sh -c 'echo "machine apt.enterprisedb.com login <username> password <password>" >
/etc/apt/auth.conf.d/edb.conf'
```

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://apt.enterprisedb.com/edb-deb.gpg.key  | sudo apt-key add -
```

5. Update the repository metadata:

```
apt-get update
```

6. Install the Debian package:

To install the latest version of PostGIS for EDB Advanced Server version 13.0:

```
apt-get install edb-as13-postgis3
```

To install the latest version of PostGIS for EDB Advanced Server version older than 13.0:

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

Where *<xx>* is the Advanced Server version older than 13.0

For example, to install the latest version of PostGIS for EDB Advanced Server version 12:

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

To install specific PostGIS version, for example 3.1.1, on Debian platforms for EDB Advanced Server version, for example 11, invoke the following commands:

On Debian 9:

```
apt-get install edb-as11-postgis-3.1=3.1.1-1+deb9
```

On Debian 10:

```
apt-get install edb-as11-postgis-3.1=3.1.1-1+deb10
```

On Ubuntu 18.04:

```
apt install edb-as11-postgis-3.1=3.1.1-1+ubuntu4
```

On Ubuntu 20.04

```
apt install edb-as11-postgis-3.1=3.1.1-1+ubuntu5
```

Installing PostGIS on a SLES 12 Host

Perform the following steps to install PostGIS on an SLES Host using the Zypper package manager:

Zypper will attempt to satisfy package dependencies as it installs a package but requires access to specific repositories not hosted at EDB.

1. Assume superuser privileges.

```
sudo su -
```

2. Use the following command to add the EDB repository to your SLES host:

```
zypper addrepo https://zypp.enterprisedb.com/suse/edb-sles.repo
```

3. Invoke the following command to refresh the metadata:

```
zypper refresh
```

4. Install `SUSEConnect` to register the host with SUSE to allow access to SUSE repositories:

```
zypper install SUSEConnect
```

5. Register the host with SUSE to allow access to SUSE repositories and replace `'REGISTRATION_CODE'` and `'EMAIL'` with your SUSE registration information:

```
SUSEConnect -r 'REGISTRATION_CODE' -e 'EMAIL'
SUSEConnect -p PackageHub/12.4/x86_64
SUSEConnect -p sle-sdk/12.4/x86_64
```

6. Install the following repository for PEM dependencies:

```
zypper addrepo
https://download.opensuse.org/repositories/Apache:/Modules/SLE_12_SP4/Apache:Module
o
```

7. Refresh the metadata:

```
zypper refresh
```

8. Install OpenJDK (version 1.8) for Java-based components:

```
zypper -n install java-1_8_0-openjdk
```

9. Use the zypper utility to install PostGIS:

```
zypper -n install edb-as12-postgis
```

Installing PostGIS on a Windows Host

You must install Advanced Server before installing PostGIS. If you have used the graphical Setup wizard to install EDB Postgres 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.

!!! Note

```
To install PostGIS version 3.1 on EDB Postgres Advanced Server version 9.6 or 10 on Windows, you need to upgrade it to the latest EDB Postgres Advanced Server minor version of 9.6.21.28 or 10.16.25 (or later), and then proceed with PostGIS 3.1 installation.
```

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.

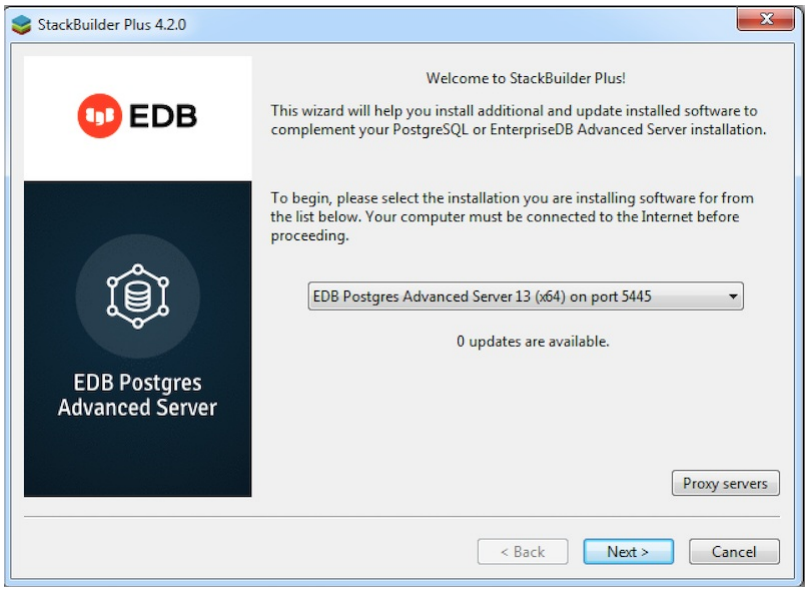

Fig. 1: The StackBuilder Plus Welcome window

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 are displayed; change the locations if required. Click `Next`.

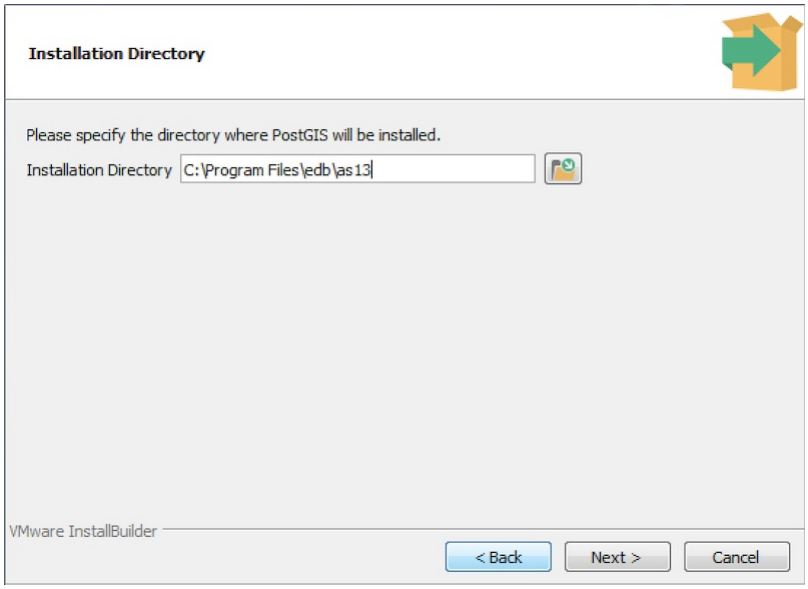

Fig. 2: The Installation Directory window

4. Once you have downloaded the installation files, a confirmation message is displayed. Click `Next` to start the PostGIS installation.

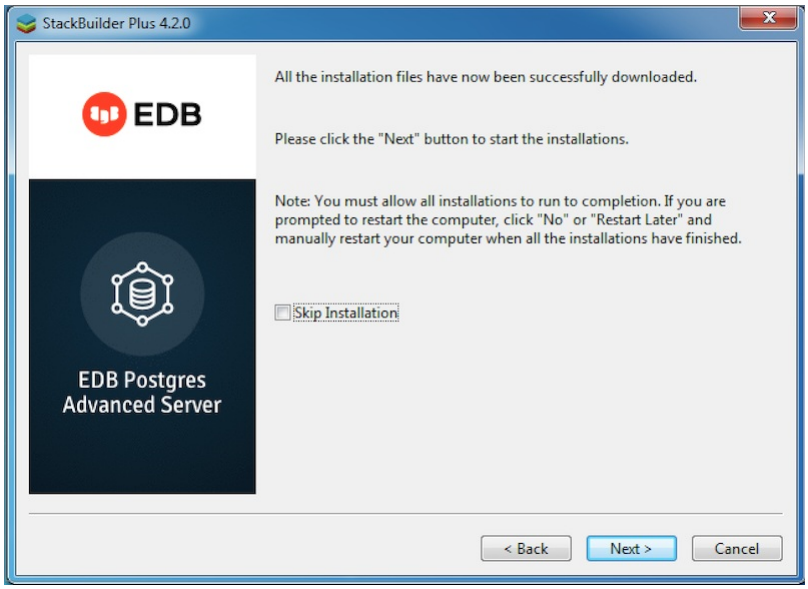

Fig. 3: Installing Postgis

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

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

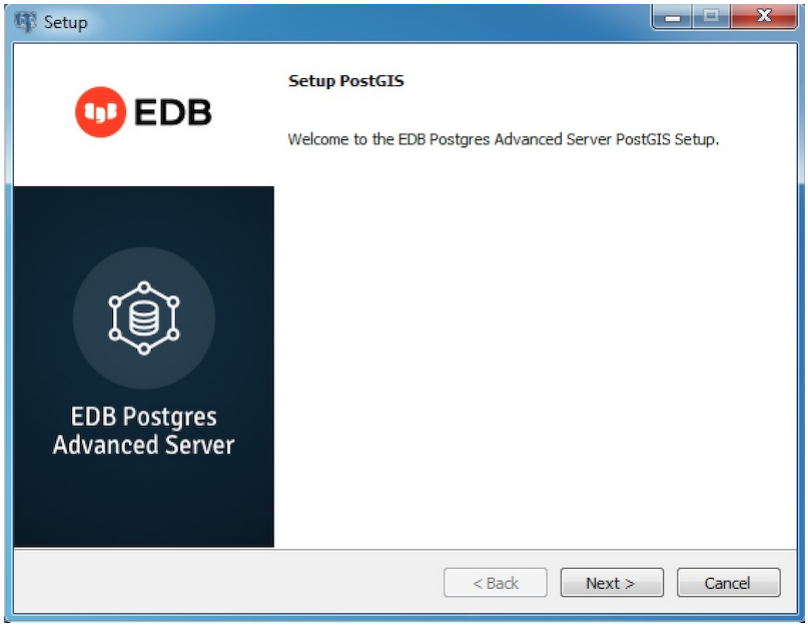

Fig. 4: The Postgis welcome window

7. Use the `Installation Directory` field to specify the directory 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.

   Click `Next` to continue.

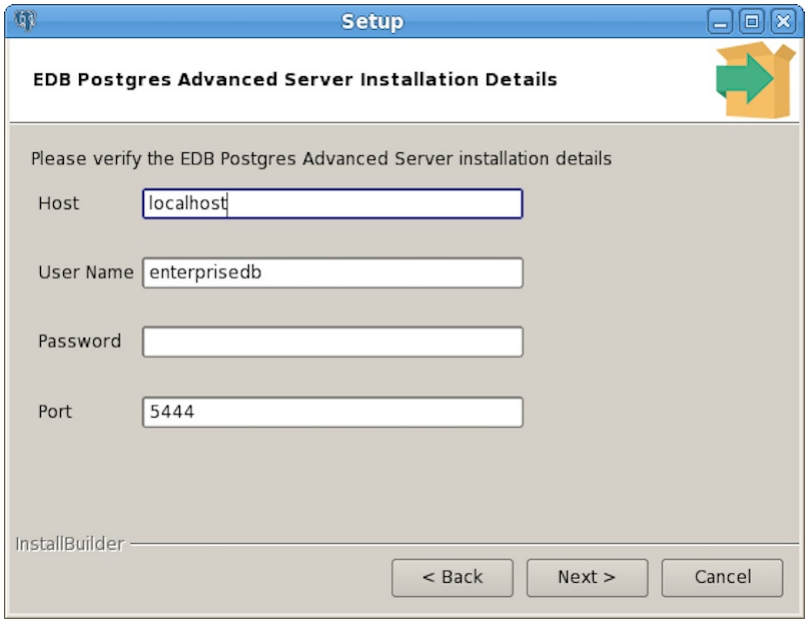

Fig. 5: The Advanced Server installation details window

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`.

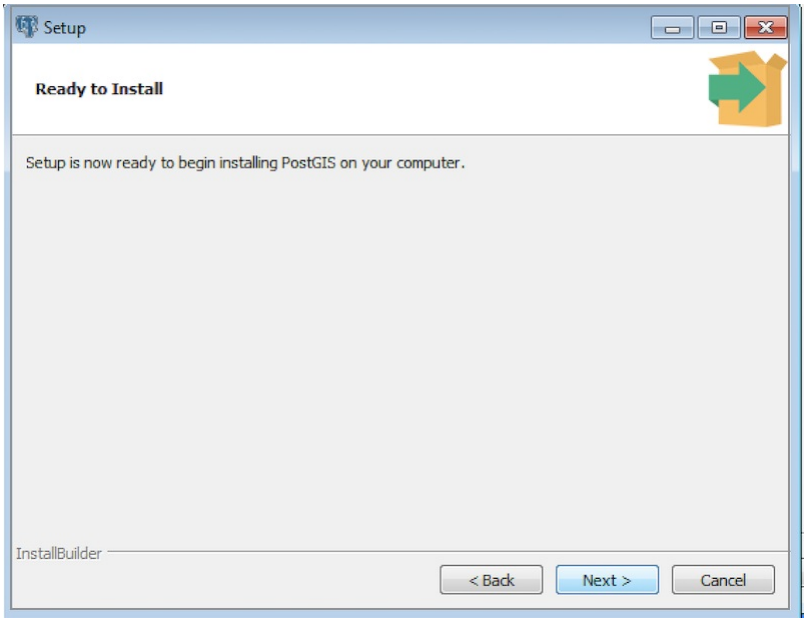

Fig. 6: The ready to install window

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

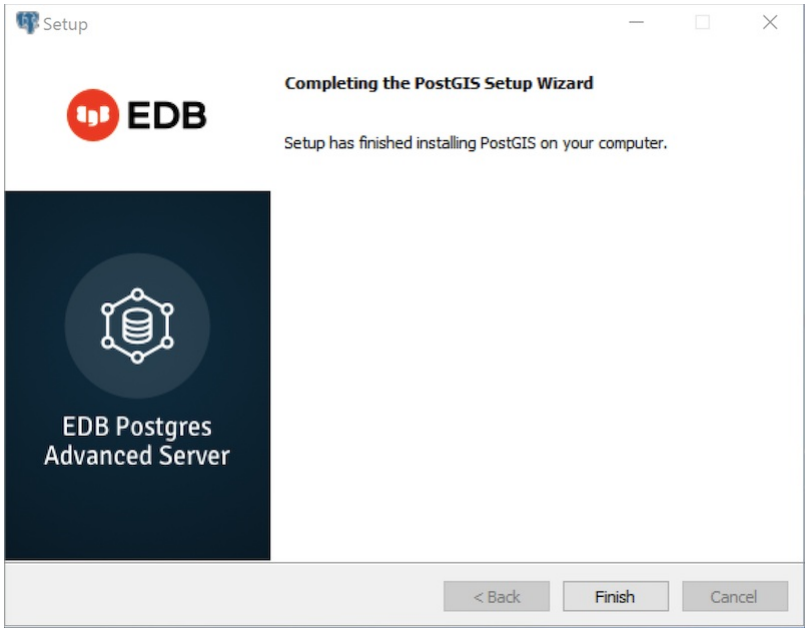

Fig. 7: The installation is complete

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

3 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. Invoke the following command to create a privileged role:

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

3. Log out of psql and connect as `gisadmin`:

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

4. Invoke the following command 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;
```

!!! Note

```
The postgis-sfcgal extension is not available on Ubuntu 18, Ubuntu 20, SLES 12,
RHEL/CentOS 7 - ppc64le, and Windows platforms.
```

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.

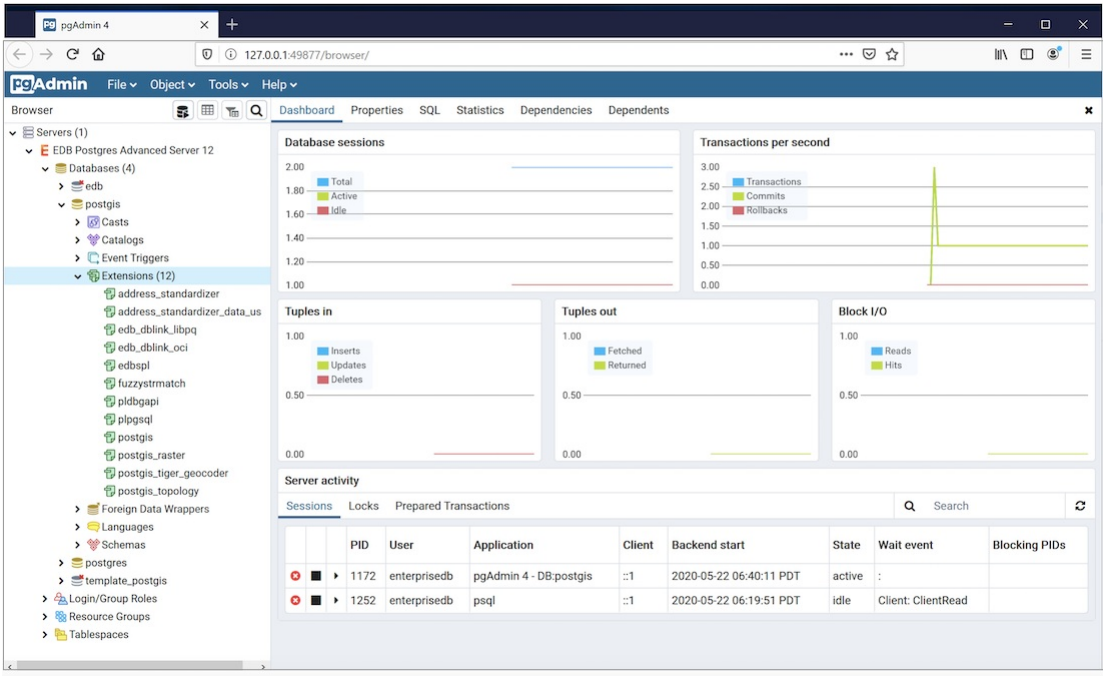

Fig. 8: Extensions on PgAdmin

4 Upgrading PostGIS

This section walks you through the following upgrade examples for RHEL/CentOS 7 platforms:

- Example 1 - Explains how to Upgrade from PostGIS version 2.4.6 to PostGIS version 3.1.1 for EDB Postgres Advanced Server version 9.6.
- Example 2 - Explains how to upgrade from PostGIS version 2.4.6 for EDB Postgres Advanced Server version 9.6 to PostGIS version 3.1.2 for EDB Postgres Advanced Server version 13.

Example 1

The following example walks you through the process of upgrading PostGIS version 2.4.6 to PostGIS version 3.1.1 for EDB Postgres Advanced Server version 9.6.

!!! Note

```
It is assumed that you have already created extensions for PostGIS version 2.4.6,
and EDB Advanced Server 9.6 service is running.
```

1. To upgrade PostGIS version 2.4.6 to PostGIS version 3.1.1 for EDB Postgres Advanced Server 9.6, assume the superuser privileges and invoke the following command:

   ```
   yum upgrade edb-as96-postgis-3.1.1 -y
   ```

2. To update extensions, switch to `enterprisedb` user, connect to the database (where you have already created extensions) with the psql client application, and execute the following commands:

   ```
   edb=# alter extension postgis update to '3.1.1';
   ```

```
WARNING: unpackaging raster
WARNING: PostGIS Raster functionality has been unpackaged
HINT: type `SELECT postgis_extensions_upgrade();` to finish the upgrade. After
upgrading, if you want to drop raster, run: DROP EXTENSION postgis_raster;
ALTER EXTENSION
```

```
edb=# alter extension address_standardizer update to '3.1.1';
ALTER EXTENSION
```

```
edb=# SELECT postgis_extensions_upgrade();
NOTICE: Updating extension postgis_sfcgal from 2.4.6 to 3.1.1
NOTICE: Packaging extension postgis_raster
NOTICE: Updating extension postgis_topology from 2.4.6 to 3.1.1
NOTICE: Updating extension postgis_tiger_geocoder from 2.4.6 to 3.1.1
          postgis_extensions_upgrade
--------------------------------------------------------------------
 Upgrade completed, run SELECT postgis_full_version(); for details
(1 row)
```

Example 2

!!! Note

```
It is assumed that you have already created extensions for PostGIS version 2.4.6,
and EDB Advanced Server 9.6 service is running.

When the PostGIS data has a dependency on the raster functions, upgrading to PostGIS
version 3.1.2 requires [dumping and reloading the data]
(https://www.enterprisedb.com/edb-docs/d/postgresql/reference/manual/13.1/app-
pgdump.html)
```

The following example walks you through the process of upgrading PostGIS version 2.4.6 for EDB Postgres Advanced Server 9.6 to PostGIS version 3.1.2 for EDB Postgres Advanced Server version 13:

- Step 1 - Upgrade to PostGIS 3.1.1 - This is an intermediate step required to resolve dependency issues.
- Step 2 - Upgrade to PostGIS 3.1.2
- Step 3 - Upgrade EDB Postgres Advanced Server version 9.6 to to 13

Step 1 - To upgrade PostGIS version 2.4.6 to 3.1.2, you need to upgrade it to 3.1.1 first.

1. Navigate to the bin directory of EDB Advanced Server 9.6:

   ```
   cd /usr/edb/as9.6/bin/
   ```

2. Assume the superuser privileges and invoke the following command to upgrade to PostGIS version 3.1.1:

   ```
   yum upgrade edb-as96-postgis-3.1.1 -y
   ```

   !!! Note See the Installing PostGIS on a Debian/Ubuntu Host section for information about Debian platform commands.

3. To update extensions, switch to `enterprisedb` user, connect to the database (where you have already created

extensions) with the psql client application, and execute the following commands:

```
edb=# alter extension postgis update to '3.1.1';
WARNING: unpackaging raster
WARNING: PostGIS Raster functionality has been unpackaged
HINT: type `SELECT postgis_extensions_upgrade();` to finish the upgrade. After
upgrading, if you want to drop raster, run: DROP EXTENSION postgis_raster;
ALTER EXTENSION
```

```
edb=# alter extension address_standardizer update to '3.1.1';
ALTER EXTENSION
```

```
edb=# SELECT postgis_extensions_upgrade();
NOTICE: Updating extension postgis_sfcgal from 2.4.6 to 3.1.1
NOTICE: Packaging extension postgis_raster
NOTICE: Updating extension postgis_topology from 2.4.6 to 3.1.1
NOTICE: Updating extension postgis_tiger_geocoder from 2.4.6 to 3.1.1
          postgis_extensions_upgrade
------------------------------------------------------------------------
 Upgrade completed, run SELECT postgis_full_version(); for details
(1 row)
```

Step 2 - Upgrade to PostGIS version 3.1.2

1. To upgrade PostGIS version 3.1.1 to 3.1.2, invoke the following command for EDB Postgres Advanced Server 9.6:

```
yum upgrade edb-as96-postgis-3.1.2 -y
```

2. To update extensions, switch to `enterprisedb` user and execute the following commands:

```
edb=# alter extension postgis update to '3.1.2';
WARNING: unpackaging raster
WARNING: PostGIS Raster functionality has been unpackaged
HINT: type `SELECT postgis_extensions_upgrade();` to finish the upgrade. After
upgrading, if you want to drop raster, run: DROP EXTENSION postgis_raster;
ALTER EXTENSION
```

With PostGIS version 3.1.2, the return type of the raster functions has changed, which requires dropping and creating the raster extension as part of the upgrade process.

```
edb=# drop extension postgis_raster;
DROP EXTENSION
```

```
edb=# SELECT postgis_extensions_upgrade();
NOTICE: Updating extension postgis_sfcgal from 3.1.1 to 3.1.2
NOTICE: Updating extension postgis_topology from 3.1.1 to 3.1.2
NOTICE: Updating extension postgis_tiger_geocoder from 3.1.1 to 3.1.2
          postgis_extensions_upgrade
------------------------------------------------------------------------
 Upgrade completed, run SELECT postgis_full_version(); for details
(1 row)
```

```
edb=# alter extension address_standardizer update to '3.1.2';
ALTER EXTENSION
```

Quit and re-open the psql client session:

```
edb=# \q
./psql -d edb -p 5444
```

```
edb=# create extension postgis_raster;
CREATE EXTENSION
```

Step 3 - Upgrade EDB Postgres Advanced Server version 9.6 to to 13:

1. Assume the superuser privileges to stop the EDB Postgres Advanced Server 9.6 service:

```
systemctl stop edb-as-9.6
```

2. Install the EDB Postgres Advanced Server version 13:

```
yum install edb-as13-server -y
```

3. Navigate to the bin directory of EDB Advanced Server 13 and initialize the cluster:

```
cd /usr/edb/as13/bin/

./edb-as-13-setup initdb
```

4. Install the PostGIS version 3.1.2 for EDB Postgres Advanced Server version 13.0:

```
yum install edb-as13-postgis3-3.1.2 -y
```

5. Assume the superuser privileges to stop the EDB Postgres Advanced Server 13.0 service:

```
systemctl stop edb-as-13
```

6. Switch to `enterprisedb` user and create a `temp` folder:

```
su enterprisedb
cd $(mktemp -d)
```

7. Execute the following commands to check cluster compatibility and consistency, as well as to perform the upgrade.

```
bash-4.2$ /usr/edb/as13/bin/pg_upgrade -d /var/lib/edb/as9.6/data/ -D
/var/lib/edb/as13/data/ -U enterprisedb -b /usr/edb/as9.6/bin/ -B
/usr/edb/as13/bin/ -p 5444 -P 5445 -c

Performing Consistency Checks
-----------------------------
Checking cluster versions ok
Checking database user is the install user ok
Checking database connection settings ok
Checking for prepared transactions ok
Checking for reg* data types in user tables ok
Checking for contrib/isn with bigint-passing mismatch ok
Checking for tables WITH OIDS ok
Checking for invalid "sql_identifier" user columns ok
Checking for invalid "unknown" user columns ok
Checking for hash indexes ok
```

```
Checking for presence of required libraries ok
Checking database user is the install user ok
Checking for prepared transactions ok
Checking for new cluster tablespace directories ok

Clusters are compatible
```

```
bash-4.2$ /usr/edb/as13/bin/pg_upgrade -d /var/lib/edb/as9.6/data/ -D
/var/lib/edb/as13/data/ -U enterprisedb -b /usr/edb/as9.6/bin/ -B
/usr/edb/as13/bin/ -p 5444 -P 5445 --link

Performing Consistency Checks
-----------------------------
Checking cluster versions ok
Checking database user is the install user ok
Checking database connection settings ok
Checking for prepared transactions ok
Checking for reg* data types in user tables ok
Checking for contrib/isn with bigint-passing mismatch ok
Checking for tables WITH OIDS ok
Checking for invalid "sql_identifier" user columns ok
Checking for invalid "unknown" user columns ok
Creating dump of global objects ok
Creating dump of database schemas
ok
Checking for presence of required libraries ok
Checking database user is the install user ok
Checking for prepared transactions ok
Checking for new cluster tablespace directories ok

If pg_upgrade fails after this point, you must re-initdb the
new cluster before continuing.

Performing Upgrade
Analyzing all rows in the new cluster ok
Freezing all rows in the new cluster ok
Deleting files from new pg_xact ok
Copying old pg_clog to new server ok
Setting next transaction ID and epoch for new cluster ok
Deleting files from new pg_multixact/offsets ok
Copying old pg_multixact/offsets to new server ok
Deleting files from new pg_multixact/members ok
Copying old pg_multixact/members to new server ok
Setting next multixact ID and offset for new cluster ok
Resetting WAL archives ok
Setting frozenxid and minmxid counters in new cluster ok
Restoring global objects in the new cluster ok
Restoring database schemas in the new cluster
ok
Adding ".old" suffix to old global/pg_control ok

If you want to start the old cluster, you will need to remove
the ".old" suffix from /var/lib/edb/as9.6/data/global/pg_control.old.
Because "link" mode was used, the old cluster cannot be safely
started once the new cluster has been started.
```

```
Linking user relation files
ok
Setting next OID for new cluster ok
Sync data directory to disk ok
Creating script to analyze new cluster ok
Creating script to delete old cluster ok
Checking for hash indexes ok

Upgrade Complete
Optimizer statistics are not transferred by pg_upgrade so,
once you start the new server, consider running:
./analyze_new_cluster.sh

Running this script will delete the old cluster's data files:
./delete_old_cluster.sh
```

8. Assume the superuser privileges, navigate to the `bin` directory of EDB Postgres Advanced Server 13, and start the service:

```
cd /usr/edb/as13/bin/

systemctl start edb-as-13
```

9. To update extensions, switch to `enterprisedb` user, connect to the database (where you have already created extensions) with the psql client application, and execute the following commands:

```
su enterprisedb
./psql -d edb -p 5444
edb=# SELECT PostGIS_Extensions_Upgrade();
NOTICE: Updating extension postgis 3.1.2
      postgis_extensions_upgrade
--------------------------------------------------------------------
Upgrade completed, run SELECT postgis_full_version(); for details
(1 row)
```

5 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) );
```

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 function 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 is created, you can use the `&&` operator 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)
```

Use the `BOX3D` function to specify a bounding box. The `&&` operator uses the index to quickly reduce the result set down to only those geometries with 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)
```

6 Uninstalling PostGIS

This section walks you through the process of uninstalling PostGIS.

Uninstalling PostGIS on a CentOS/RHEL Host

To uninstall PostGIS on a CentOS/RHEL host, assume the identity of the root user and invoke the following command:

On CentOS/RHEL 7 for EPAS version 13:

```
yum -y erase edb-as13-postgis3*
```

On CentOS/RHEL 7 for older versions of EPAS:

```
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.

On CentOS/RHEL 8 for EPAS version 13:

```
dnf -y erase edb-as13-postgis3*
```

On CentOS/RHEL 8 for older vesions of EPAS:

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

Uninstalling PostGIS on a Debian/Ubuntu Host

To uninstall PostGIS on a Debian or Ubuntu host, 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 SLES Host

To uninstall PostGIS on a SLES host, assume the identity of the root user and invoke the following command:

```
zypper remove edb-as12-postgis*
```

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\as13`).

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

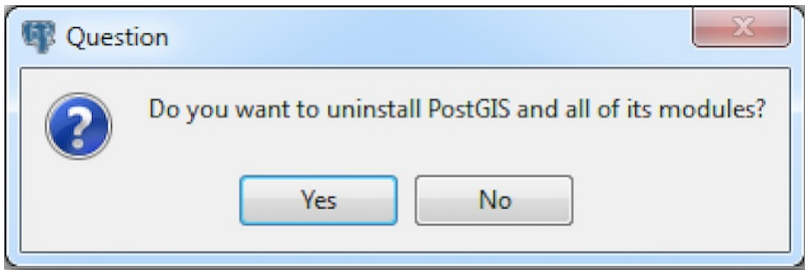

Fig. 9: The Uninstaller Confirmation Dialog

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

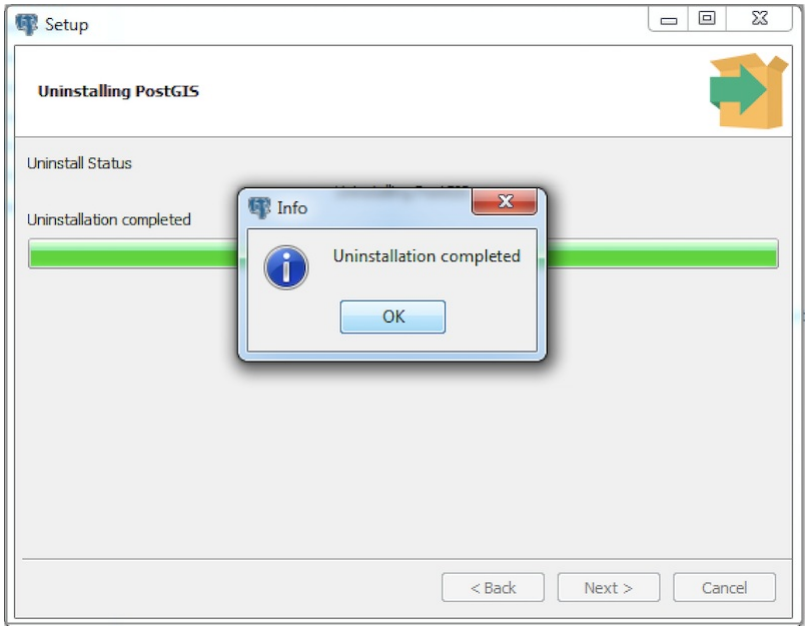

Fig. 10: Uninstallation is Complete